Mission Tool

Design Review

November 19, 2001

Agenda

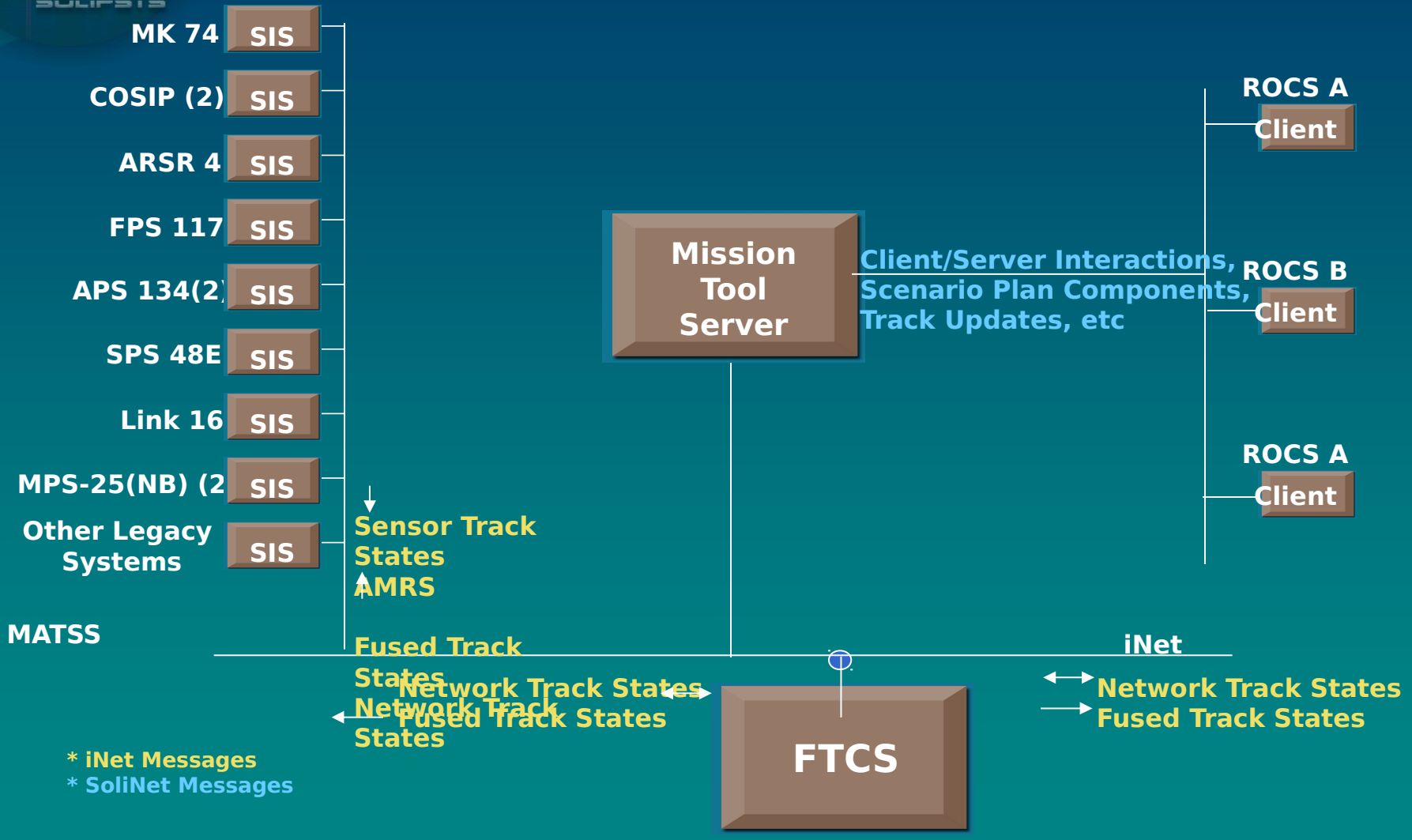
- **System Architecture**
- **Current Capabilities**
- **Software Development Process**
- Meta-Compiler System
- **Software Development Plan**
- **New Requirements**

© 2001 Solipsys Corporation

System Architecture

© 2001 Solipsys Corporation

ESPRIT Architecture



© 2001 Solipsys Corporation

Range Mission Tool/TDF Relationship

RMT GUI is a set of JAVA plugins built on top of the Tactical Display Framework

- **Range Mission Tool Client GUI built upon TDF**
  - **Many features for “free”**
  - **Code base used over many projects**
    - **Supports stability**
    - **Broader user base increases likelihood of bug discovery/resolution**
- **Interfaces**
  - **Allows interaction with classes without specific knowledge of individual class**
- **Application base classes allow rapid development**
  - **No re-inventing the wheel for basic application mechanics**
- **Plugins**
  - **No changes to code base**
  - **Plugin mechanism allows new menu items, track interfaces, view objects, etc. to be inserted dynamically**

© 2001 Solipsys Corporation

Interface Based System

- **Many components used in system are interface driven**
  - **Allows flexibility and extensibility**
  - **Examples:**
    - **Track -- a tagging interface, i.e., any class can be a track**
    - **ViewPointer -- pointers with different behaviors can be swapped in**
    - **ClientModules -- allow access to the track database from various sources**
  - **Listener Mechanism**
    - **Allows custom components to get notified when events occur**

© 2001 Solipsys Corporation

TDF Base Class Examples

- **Application**
  - Manages common functionality in sub-applications
    - DocumentApplication
    - FileApplication
- **BasicClientModule**
  - Manages common functionality in track interfaces
    - EspritClientModule
    - MSCTClientModule
- **AbstractModel**
  - Base class for data models
  - Allows multiple views independent of data

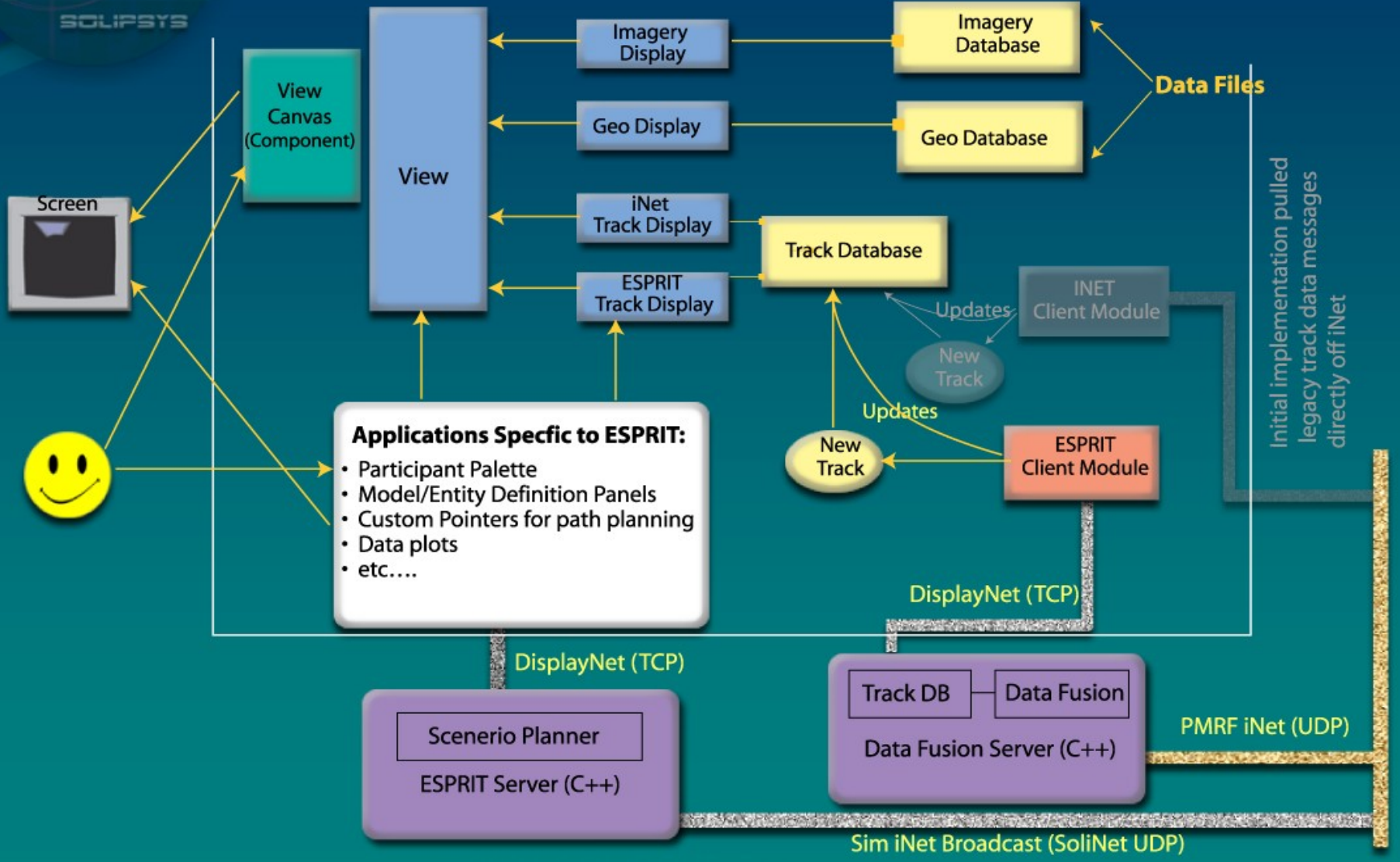
© 2001 Solipsys Corporation

Range Mission Tool/TDF Plugins

- **How do Plugins actually work?**
  - **Any class can be a Plugin**
  - **Upon startup, DynamicLoader loads Plugin classes listed in Plugin lists**
  - **Applications request all Plugins of a certain class from the DynamicLoader**
  - **Plugin applications can in turn load their own Plugins**
    - **For example, Preferences Application (a Plugin) dynamically loads PreferencePanels, which in turn may add Plugin components of their own**

© 2001 Solipsys Corporation

Range Mission Tool/TDF Relationship



Current Capabilities

© 2001 Solipsys Corporation

Current Capabilities

- Pre-Planning
- Planning
- Playback
  - Preview
  - Rehearsal
- Post Op

© 2001 Solipsys Corporation

Preplanning

- **Sensor model entry**
- **Vehicle model entry**
- **Entity definition**
- **Data set import**
- **Data set creation**
- **Overlay generation**

© 2001 Solipsys Corporation

Scenario Planning

- Vehicle entity assignments
  - Data sets
    - Imported
  - Waypoint definition
- Sensor entity assignment
  - Precision vs. Surveillance
  - Sensor coverage analysis
- Overlays
- Event timeline synchronization

© 2001 Solipsys Corporation

Scenario Planning:
Planning Aids

- Planning reports
- Radar coverage analysis
- Validity checks
- Time tics and synchronization lines
- Preview mode
- Multi-station playback

© 2001 Solipsys Corporation

Event Rehearsal

- **Event replay modes:**
  - **Preview**
  - **Multi-station playback**
  - **Extended execution (planned)**

© 2001 Solipsys Corporation

Event Monitor

- **Visual aids to:**
  - **Monitor real-time planned vs. actual position and path**
  - **Overlays used to delineate and monitor safety constraints**
  - **Visual aids provided to monitor:**
    - **Trajectory**
    - **Instantaneous Intercept Points (IIPs)**
    - **Aspect angle to shooter**
    - **Track analysis plots**

© 2001 Solipsys Corporation

Use Case Overview for Baseline V3.1

- Pre-planning
  - Define a Sensor Model
  - Define a Sensor Entity
  - Define a Vehicle Model
  - Define a TBM Model based on a “Duck” Dataset
- Scenario planning
  - Create a scenario
    - Add a TBM vehicle entity that uses the TBM model
    - Add a vehicle entity based on a non-TBM vehicle model and define its trajectory
    - Synchronize vehicle paths
    - Set T0
    - Set start times for non-synchronized paths
    - Place a sensor and assign sensor-tracking responsibility
    - Check signal-to-noise for a vehicle trajectory based on sensor assignments
    - Generate vehicle reports for use during the mission
  - Save a scenario
  - Certify a scenario (via a non-secure logon to server)

© 2001 Solipsys Corporation

Use Case Overview for Baseline V3.1 (cont'd)

- **Scenario execution**
  - **Preview a scenario**
  - **Rehearse a scenario using simulated radar tracking**
  - **Monitor real-time events against planned scenario and record for later playback**

© 2001 Solipsys Corporation

Use Case Analysis: GUI-Server Message Exchange

Pre-planning use cases

- **Define a sensor model/ sensor entity/ vehicle model**
  - **Save as working**
    - **No GUI-Server messages exchanged because working models and entities are saved on client**
  - **Save as certified**
    - **← sends GUIScenarioObject**
    - **→ receives SrvScenarioObjectResponse**
- **Define a TBM model based on “Duck” dataset**
  - **← sends GUIConvertTextModel**
  - **→ receives SrvDataSetConversionCompleted**

© 2001 Solipsys Corporation

Use Case Analysis: GUI-Server Message Exchange

Scenario planning use cases

- **Create a Scenario**
  - ← sends GUIScenarioControl, action = new scenario
  - → receives SrvScenarioControlResponse, action = new scenario, status = success
  - **Add a TBM Vehicle Entity and Trajectory that uses the TBM model**
    - ← sends GUIVehicle, action = new
    - → receives SrvPlannedPathDump…
      - (for each waypoint on planned trajectory)
    - → receives SrvTimeTic
  - **Add Synch Points for TBM Trajectory**
    - ← sends GUIVehicleAnnotation…
      - (for each synch point)

© 2001 Solipsys Corporation

Use Case Analysis: GUI-Server Message Exchange

Scenario planning use cases

- **Create a Scenario (cont'd)**
  - **Add a Vehicle Entity based on a Non-TBM Vehicle Model and define its trajectory**

⟵ sends GUIVehicleModel, action = new
⟵ sends GUIVehicle, action = new

⟵ sends GUIWaypointPosition, action = first
⟶ receives SrvWaypointPath
⟶ receives SrvWaypointPosition

- **For each point added via the GUI:**

⟵ sends GUIWaypointPosition, action = successor
⟶ receives SrvWaypointPath…
(for each path segment)
⟶ receives SrvWaypointPosition…
(for each waypoint)

⟶ receives SrvTimeTic

© 2001 Solipsys Corporation

Use Case Analysis: GUI-Server Message Exchange

Scenario planning use cases

- **Create a Scenario (cont'd)**
  - **Delete Initial/ Following Waypoint**
    - ← sends GUIWaypointPosition, action = delete
    - → receives SrvWaypointPath…
      - (for each path segment)
    - → receives SrvWaypointPosition…
      - (for each remaining waypoint)
  - **Extend Path Forward/ Insert Succeeding Waypoint**
    - ← sends GUIWaypointPosition, action = successor
    - → receives SrvWaypointPath…
      - (for each path segment)
    - → receives SrvWaypointPosition…
      - (for each waypoint)

© 2001 Solipsys Corporation

Use Case Analysis: GUI-Server Message Exchange

Scenario planning use cases

- **Create a Scenario (cont'd)**
  - **Extend Path Backward/ Insert Preceding Waypoint**
    - ← **sends GUIWaypointPosition, action = predecessor**
    - → **receives SrvWaypointPath…**
      - **(for each path segment)**
    - → **receives SrvWaypointPosition…**
      - **(for each waypoint)**
  - **Modify Existing Waypoint (by dragging or via InfoPanel)**
    - ← **sends GUIWaypointPosition, action = move**
    - → **receives SrvWaypointPath…**
      - **(for each path segment)**
    - → **receives SrvWaypointPosition…**
      - **(for each waypoint)**

© 2001 Solipsys Corporation

Use Case Analysis: GUI-Server Message Exchange

Scenario planning use cases

- **Create a Scenario (cont'd)**
  - **Synchronize Paths**
    - ← sends GUIVehicleSynch
    - → receives SrvPlannedPathDump…
      (for each waypoint on trajectory)
    - → receives SrvVehicleAnnotation…
      (for each synch point)
    - → receives SrvVehicleSynch, action = add
  - **Set T0**
    - ← sends GUITicReferencePoint
    - → receives SrvTimeTic…
      (for each vehicle synchronized to that tic point)
  - **Set Start Times of Non-Synchronized Paths**
    - ← sends GUIVehicleModel, action = update
    - ← sends GUIVehicle, action = update
    - ← sends GUIVehicleStartTime, action = update
    - → receives SrvWaypointPosition…
      (for each waypoint to be synchronized)

© 2001 Solipsys Corporation

Use Case Analysis: GUI-Server Message Exchange

Scenario planning use cases

- **Create a Scenario (cont'd)**
  - **Place a Sensor and assign Sensor-tracking responsibility**
    - ← **sends GUISensorModel, action = new**
    - ← **sends GUISensor, action = new**
    - ← **sends GUISensorModeAssignment**
  - **Check Signal-to-Noise for Vehicle Trajectory based upon Sensor Assignments**
    - ← **sends GUIRequestSNvsTime**
    - → **receives SrvSNvsTimeResponse**
  - **Request Vehicle Report**
    - ← **sends GuiRequestVehicleReport**
    - → **receives SrvVehicleReportCompleted**

© 2001 Solipsys Corporation

Use Case Analysis: GUI-Server Message Exchange

Scenario planning use cases

- **Save Scenario**
  - ← sends GUIScenarioObject
  - ← sends GUIScenarioControl, action = save working scenario
  - → receives SrvScenarioObjectResponse
  - → receives SrvScenarioControlResponse
- **Close Scenario**
  - ← sends GUIScenarioControl, action = new scenario
  - → receives SrvScenarioControlResponse, action = new scenario, status = success
  - Server sends messages to remove scenario information from the display:
  - → receives GuiVehicle, action = delete,

    GuiSensor, action = delete, etc.

© 2001 Solipsys Corporation

Use Case Analysis: GUI-Server Message Exchange

Scenario planning use cases

- **Certify Scenario**
  - **Logon to Server (non-secure)**
    - ← **sends GuiCertifiedPassword, action = Request**
    - → **receives SrvCertifiedPasswordResponse**
    - ← **sends GuiScenarioObject**
    - ← **sends GuiScenarioControl, action = SaveCertifiedScenario**
    - → **receives SrvScenarioObjectResponse**
    - → **receives SrvScenarioControlResponse, action = SaveCertifiedScenario, status = Success**

© 2001 Solipsys Corporation

Use Case Analysis: GUI-Server Message Exchange

Scenario execution use cases

- **Open Scenario**
  - ← **sends GUIScenarioControl, action = load workin scenario**
  - → **receives SrvScenarioControlResponse, action = load working scenario, status = start**
    - **Server sends scenario, vehicle, sensor, etc. info, e.g. the following for a TBM trajectory:**
    - → **receives GUIVehicleModel**
    - → **receives GUIVehicle**
    - **receives SrvPlannedPathDump…**
    - → **(for each point on the trajectory)**
    - **receives GUIVehicleAnnotation**
    - →
    - → **receives GUIScenarioObject**
    - **receives SrvScenarioControlResponse, action = load working scenario, status = success**
    - → **receives SrvScenarioObjectResponse**

© 2001 Solipsys Corporation

Use Case Analysis: GUI-Server Message Exchange

Scenario execution use cases

- **Preview Scenario**
  - **Open Scenario**
  - **Select Playback Mode = Preview**
    - **[NONE]**
  - **Initiate a Scenario Timeline by specifying a Scenario time to start**

    sends GUIScenarioControl, action = set scenario start

    receives SrvScenarioControlResponse, action = set scenario start, status = success ←
  - **Play Scenario** →

    sends GUIScenarioControl, action = set clock rate →

    receives SrvScenarioControlResponse, action = set clock rate, status = success ←

    sends GUIScenarioControl, action = start preview →

    receives SrvScenarioControlResponse, action = start preview, status = success ←

    receives SrvScenarioSynch →

    receives SrvPlannedTrackState messages →

© 2001 Solipsys Corporation

Use Case Analysis: GUI-Server Message Exchange

Scenario execution use cases

- **Preview Scenario (cont'd)**
  - **Hook Planned Position**
    - **[No GUI-Server messages – this is a TDF function]**
  - **Display Alt Graph**
    - **[No GUI-Server messages – this is a TDF function]**
  - **Stop Scenario**

    ← **sends GUIScenarioControl, action = stop**

    → **receives SrvScenarioControlResponse, action = stop, status = success**

© 2001 Solipsys Corporation

Use Case Analysis: GUI-Server Message Exchange

Scenario execution use cases

- **Rehearse Scenario using Simulated Radar Tracking**
  - **Open Scenario**
  - **Activate Track Simulation using iNet Message 25**
    - ← **sends GUIScenarioControl, action = set sim type**
    - → **receives SrvScenarioControlResponse, action = set sim type, status = success**
  - **Initiate a Scenario Timeline by specifying a Scenario Time to start**

© 2001 Solipsys Corporation

Use Case Analysis: GUI-Server Message Exchange

Scenario execution use cases

- **Rehearse Scenario using Simulated Radar Tracking (cont'd)**
  - **Play Scenario**
    - ← **sends GUIScenarioControl, action = set clock rate**
    - → **receives SrvScenarioControlResponse, action = set clock rate, status = success**
    - ← **sends GUIScenarioControl, action = start local execution**
    - → **receives SrvScenarioControlResponse, action = start local execution, status = success**
    - → **receives SrvScenarioSynch**
      - **Server sends iNet messages that are processed by Data Fusion and transformed into TrackUpdate**
    - → **messages**
    - → **receives TrackUpdate messages**
    - → **receives SrvDisplayTrackState messages**
    - **receives SrvPlannedTrackState messages**
  - **Hook Planned Position**
  - **Display Alt Graph**
  - **Stop Scenario**

© 2001 Solipsys Corporation

Use Case Analysis: GUI-Server Message Exchange

Scenario execution use cases

- **Monitor Real-time Events Against Planned Scenario and Record for later Playback**
  - **Data Fusion processes iNet messages MT25 & MT 26 and transforms them into TrackUpdate msgs**

    → receives TrackUpdate
  - **Open Scenario**
  - **Synchronize Scenario Timeline to Mission Countdown Clock**

    ← sends GUIScenarioControl, action = set scenario start

    → receives SrvScenarioControlResponse, action = set scenario start, status = success
  - **Record using the Track Recorder**
    - **[No GUI-Server messages – this is a TDF function]**
  - **Play Scenario**
  - **Display Alt Graph**

© 2001 Solipsys Corporation

Current Status

- FY ’99 Acceptance Test in September 1999:
  - Included Transition Plan
  - Noted deficiencies repaired for Dec ’99 delivery
- FY ’00 funding for:
  - Operational assessment during three events
  - Limited funds available for enhancements
- Current FY ’01 funding for:
  - Addition of 14 new required features
  - Event support

© 2001 Solipsys Corporation

Software Development Process

© 2001 Solipsys Corporation

Dispelling Popular Myths

- **We are merely prototyping**
- **We are rogue hackers from Hell**
- **We are lacking process**

The more things change - the more things stay the same.

© 2001 Solipsys Corporation

Instilling Actual Fact

Hell wouldn't have us!

- **Intimately familiar with rigorous development techniques**
- **Bring many years of experience to solving "real world" issues**
- **Dynamic nature of staff allows for rapid turnaround of viable and robust solutions - often confused with "prototyping"**
- **We do not get bogged down by process**

© 2001 Solipsys Corporation

The High Level View

Standard approaches are taken to manage, track, and maintain the various products and projects.

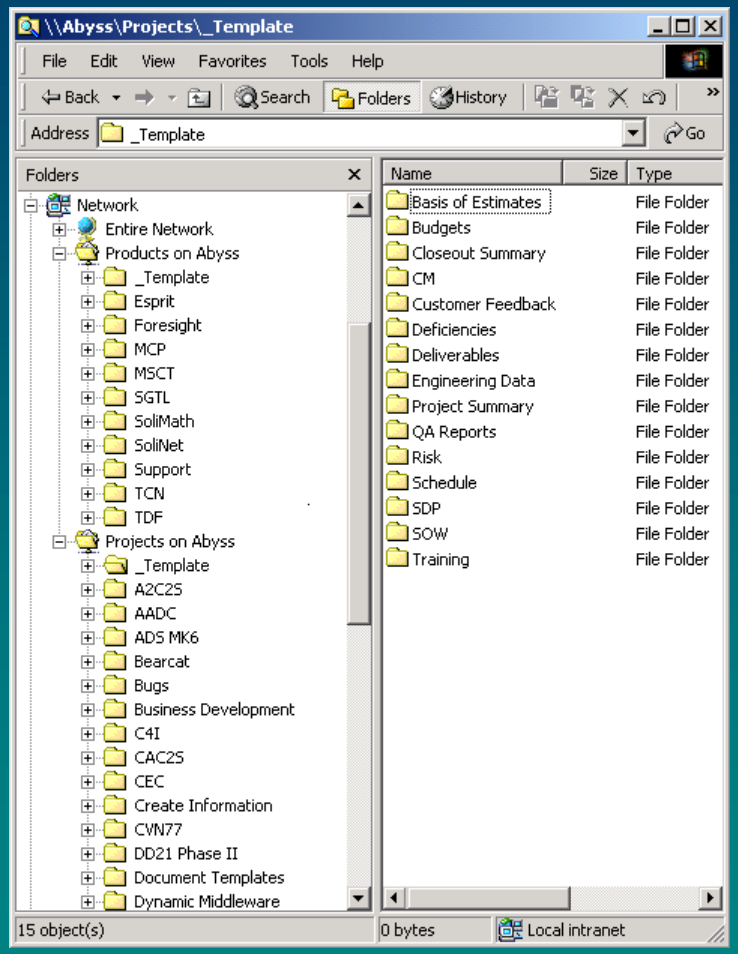
- **Product and project documentation is created and stored on a networked server**
  - **Access to each folder is controlled by product or project Manager**
  - **Each folder is accessible to the QA Department for auditing purposes**
- ***Proceed* Tool promotes a uniform structure across folders**
  - **Allows QA Team to play a less intrusive role which makes everyone happy**
  - **Automated updates to push information to customer support area**
- **Development suite is comprised of standard GNU toolset and custom scripts**

© 2001 Solipsys Corporation

View Into the Abyss

- **Primary server is named Abyss**
- **Provides network access to all product and project folders**
- **System is monitored and maintained by system admin personnel**
- **Supports the Solipsys Proceed Tool**

The product and project folders reside on Abyss, files are backed up nightly, and off site storage is rotated weekly.



© 2001 Solipsys Corporation

Process Control with an Electronic Edge (Proceed)

Process and QA are terms that are typically met with much resistance by those who view it as "the Man just beating me down."

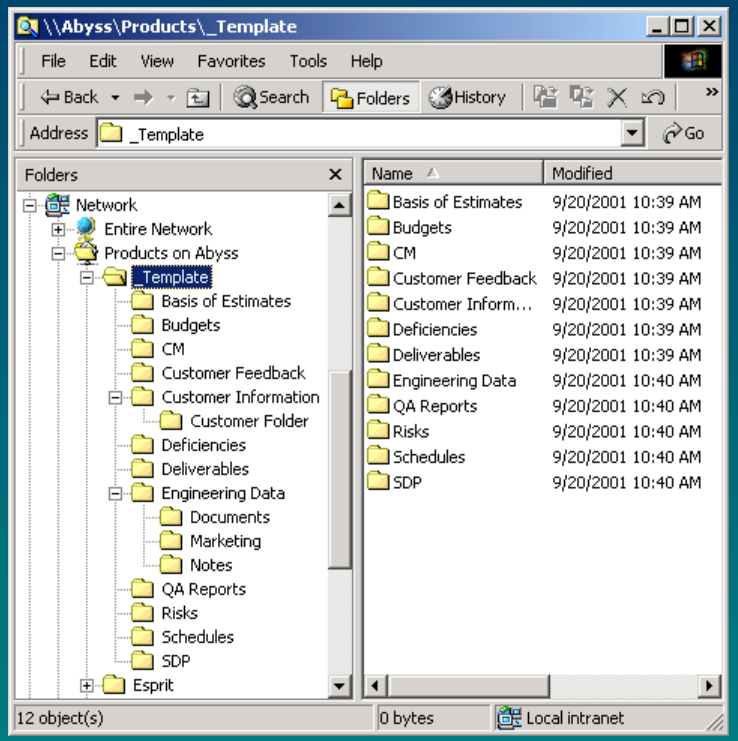
- **Manages the directory hierarchies in the products and projects areas**
  - **Applies a directory structure template to all areas managed by Proceed**
  - **Performs consistency checks to ensure that the area is not being polluted**
  - **Does not inhibit, only guides**
- **Pushes desired files from configuration area to customer support web site**
  - **Managers may selectively allow any file in the hierarchy to propagate to the web site**
  - **Proceed will automatically update the files on the web site on a regular basis**
- **QA can play a more active role without upsetting the "Volatiles"**

© 2001 Solipsys Corporation

Proceed Hierarchy Template

Proceed serves as an enabler to allow QA to lend a hand with maintaining a consistent and recognizable structure in a less invasive manner.

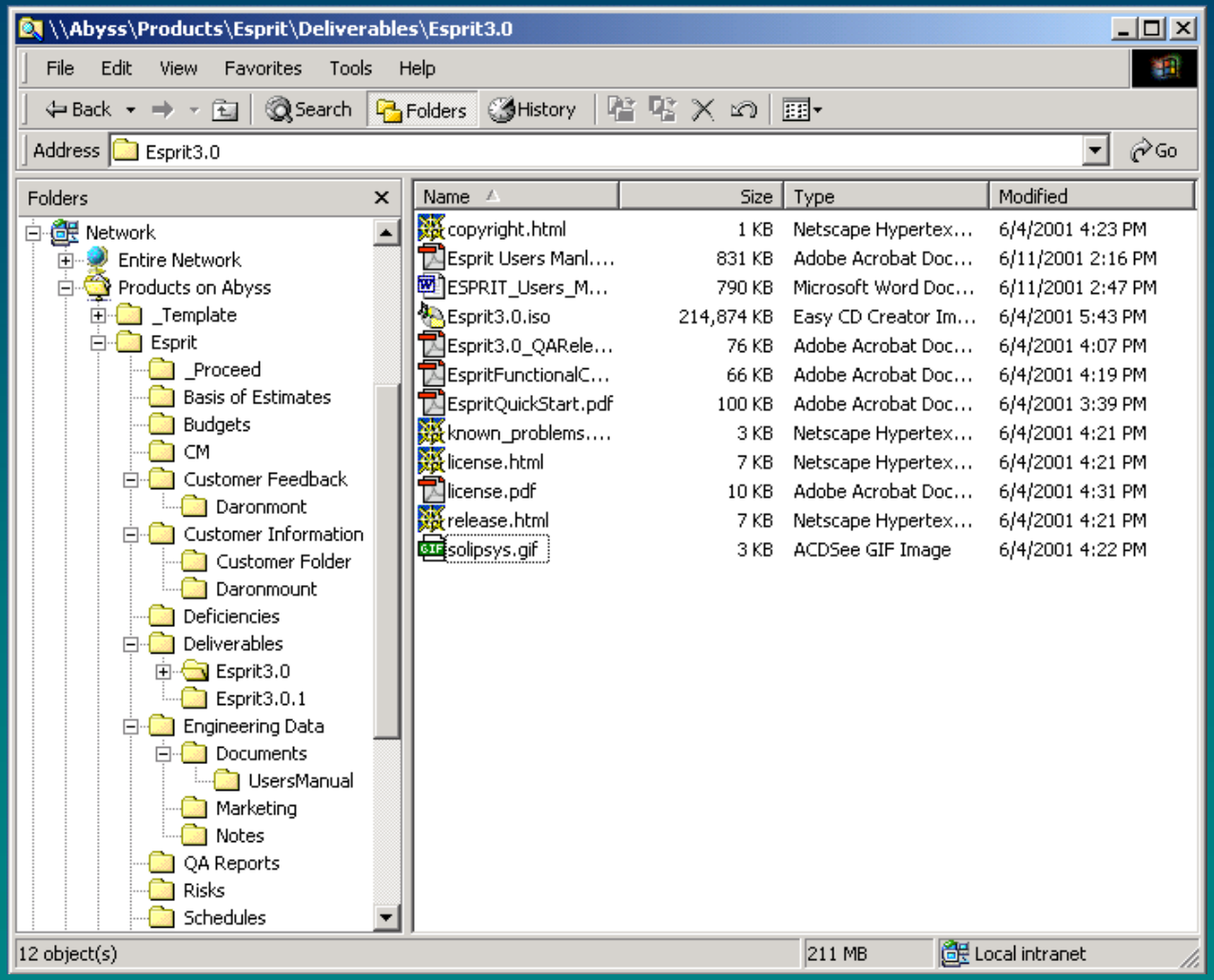
- **The QA Team maintains the hierarchy template**
- **Proceed ensures hierarchical consistency across all products and projects folders**



© 2001 Solipsys Corporation

Sample Product Folder - Esprit

The Esprit product folder contains deliverable ISO images, licensing information, QA release notes, bug lists, etc.



© 2001 Solipsys Corporation

Software Development Environment

The GNU Toolset, automation of process, and reusable software components are the foundation of the software development cycle.

- **Standard GNU Toolset is employed**
  - **Available for most platforms**
  - **Promotes portability**
  - **Widely supported by the Net community**
  - **Cost effective**
- **Custom in-house techniques**
  - **Scold**
  - **Rebuild**
  - **Metagen and Cajun**
- **Solid foundations through software reuse**
  - **SoliNet**
  - **SoliMath**
  - **Slate**
  - **TDF**

© 2001 Solipsys Corporation

GNU Tool: Concurrent Versions System (cvs)

- **Network-based Revision Control System**
- **Enables multiple developers to work on parts of the system**
  - **Reduces amount of “dead time” caused by a lock based system**
  - **May introduce conflicts if the exact same area is modified, but CVS supports conflict reporting and resolution**
- **Local environment is replicated on the server at PMRF**
  - **Developers enjoy a familiar environment when in the field**
  - **Necessary for distributed testing and development**

CVS is based on older Revision Control System (RCS); both are commonly used GNU packages.

© 2001 Solipsys Corporation

GNU Tool: CVS *(cont'd)*

Concurrency, consistency, and conformity are our cornerstones to building and maintaining robust systems.

- **Supports branching for use when configurations are taken into the field**
- **Baseline configuration project (CVSproject) is used as the genesis for all new projects**
  - **Ensures a consistent hierarchy across all projects in the system**
  - **Allows new projects to start with a robust and well tested configuration**
  - **The CVSproject is maintained and kept current**
- **Available for Solaris (Sparc/x86), various flavors of Linux, and Cygwin32**

© 2001 Solipsys Corporation

GNU Tool: CVS (cont'd)

- **Maintains status logs and modification commentary**
- **May be configured to use custom scripts for automated generation of change reports**

The sample change log is from the original mods to the SoliNet make and auto configuration files.

```
==========================================
===============
Root: /thor/CVS                    User: john
08/18/98
Dir : solinet
15:18:41
File:  Makedefs.in Makefile.in Makerules.in
acsite.m4 archive
      configure configure.in  Mods:
  **   Massive changes went in to support a software
  **   installation. Most of the modifications were in
support
  **   of placing the SoliNet and SoliMessage
include files down
  **   one level rather then having everything in
/include. The
  **   changes to the make system were made in
support of adding
  **   SoliNet to the scold automated source code
checking
  **   scheme. All of the modifications made to the
make system
  **   were migrated to the cvsproject template
directory which
  **   is maintained under CVS control.
==========================================
===============
Root: /thor/CVS                    User: john
08/18/98
Dir : solinet
15:32:06
File:  configure configure.in  Mods:
  **   Restructured the src/lib directories by
renaming solinet
  **   to SoliNet, solimsg to SoliMessage, and
metagen to
  **   MetaGen.
==========================================
===============
```

© 2001 Solipsys Corpor

GNU Tool: autoconf

Using autoconf and SoliNet completely divorces the software from the system.

- **Not all Unixes are created equal**
  - **Ferreting out subtle differences between systems can be a configuration nightmare**
  - **Developing and maintaining portable scripts is a monumental task**
- **Autoconf allows configure.in files to specify high-level tests**
  - **The configure.in file is parsed and a configure script is generated using Bourne shell syntax**
  - **The resultant script can be executed on any Unix platform**
  - **The test results are applied to the make system**
- **SoliNet utilizes autoconf to completely isolate system dependencies from the application**

© 2001 Solipsys Corporation

GNU Tool: make

Make is responsible for the compiler and linker interactions necessary to create executable code from the system's source code.

- **Make is actually a Unix utility which GNU has extended**
  - **Maintains dependencies between files**
  - **Understands how to rebuild pieces of the system based on an extensible rule set**
  - **Rules exist to utilize Solipsys meta language extensions**
- **Recursively drills through a project's directory hierarchy building the system based on the rules**
- **Employed by CVS repositories for rebuilding the world**

© 2001 Solipsys Corporation

GNU Tool: gcc

gcc is a fully ANSI compliant C/C++ compiler that is available on a wide range of architectures.

- **ANSI compliant C++ compiler**
- **Used in nit pick mode**
  - **Full type checking is performed**
  - **All non-standard constructs are flagged as warnings**
  - **All unsafe operations are flagged as warnings**
  - **All warnings are treated as errors and may not be ignored**
- **Make and configuration system controls the use of *gcc***
  - **Users cannot disable the warning level imposed by the system**
  - **Users must fix all warnings and errors before a functional system may be completely built**

© 2001 Solipsys Corporation

In-House Tool: Scold

Scold acts as a source code and configuration watchdog by performing auto-builds on a regular basis.

- **The Source Code OverLordD (Scold) is a configuration area as well as a set of scripts**
- **Used as the central repository for latest builds**
  - **Obviates the need for each developer to build local copies of the most commonly used toolkits**
  - **Maintained by the configuration manager**
- **Also refers to the automated Scold process**
  - **Perform periodic checkout and build of the systems**
  - **Performs analysis of the logs from the build process**
  - **Flags problems and notifies last person that performed a modification to the problem unit**

© 2001 Solipsys Corporation

In-House Tool: Rebuild

The rebuild script allows the build manager to easily build the configured software on any machine.

- **Perl script that understands the package build order**
- **Provides a simple one line command to build any or all of the systems**
  - **Can rebuild the world from scratch**
  - **Can update and build only modified systems**
  - **Allows the build manager to selectively create specific versions of the systems**
- **Creates a complete log of the build process as well as system specific breakdowns of what took place**

© 2001 Solipsys Corporation

In-House Tool: Metagen and Cajun

Use of the meta languages allows the machines to do the dirty work!

- **Messages and classes are described in a simple high-level language**
- **Description files pass through meta-language compiler to produce complete C++ and Java classes**
- **Class source code is then processed by the C++ and Java compilers to create fully functional objects**
  - **Eliminates common mistakes in defining and maintaining message structures**
  - **Insures consistent APIs between all C++ and Java components within the system**

© 2001 Solipsys Corporation

Software Reuse: SoliNet

SoliNet enforces strong typing and encapsulation of data, thus promoting solid object oriented designs.

- **An ANSI C++ compliant class library initially created to provide a wrapper for commonly used network interfaces**
- **Took on the role of being the operating system layer used by emerging Solipsys technologies**
  - **SoliMath**
  - **Slate/Granite**
  - **DataFusion**
  - **RMT**
  - **TCN**
- **Evolved Into a lightweight, portable operating system layer with extensive support for distributed heterogeneous communications**

© 2001 Solipsys Corporation

Software Reuse: SoliNet
(cont)

SoliNet is at the heart of the software being deployed at PMRF.

- **Foundation for the RMT Server**
- **Provides logical building blocks and network infrastructure for the range upgrade effort**
  - **Bridges the PMRF iNet to the Solipsys kNet through the FTCS**
  - **Foundation for the Source Integration Servers**

© 2001 Solipsys Corporation

Software Reuse: SoliMath

Common math classes like filters, state estimation, covariances, bias estimation, bias measurement, and more reside in SoliMath.

- **Large collection of math specific classes**
  - **Well over 100 classes currently reside in the toolkit**
  - **Functionality of classes backed by MatLab models developed and tested by analysis staff**
  - **Utilizes SoliNet as foundation therefore instances of SoliMath objects can be passed in messages**
- **Utilized heavily by other Solipsys projects and products**
  - **Slate/Granite**
  - **DataFusion**
  - **RMT**
  - **TCN**
  - **JLENS**

© 2001 Solipsys Corporation

Software Reuse: Slate

Solipsys' Latest Attempt to Track Everything

- **Slate is a software library of re-useable Tracker components**
  - **Includes classes for track initiation, track-to-contact association, activity control, promotion, and disclosure.**
  - **Application independent through parameterization and the ObserverThread.**
  - **Extensive use of SoliMath and SoliNet**

© 2001 Solipsys Corporation

Software Reuse: Tactical Display Framework (TDF)

Use of Java technology and Object Oriented Design enables TDF to swiftly take on any new display role thrown at it.

- **Foundation for all Solipsys graphical user interfaces (GUI)**
- **Easily tailored by use of Java plug-In technology**
- **Basis for several different display products at PMRF**
  - **Mk74 SIS**
  - **CoSip SIS**
  - **Surveillance SIS**
  - **RMT**
- **Presents a common look and feel across the various operator displays**

© 2001 Solipsys Corporation

Meta-Compiler System

© 2001 Solipsys Corporation

Meta-Compiler in Action

Building a class with the meta language is fast, easy and correct.

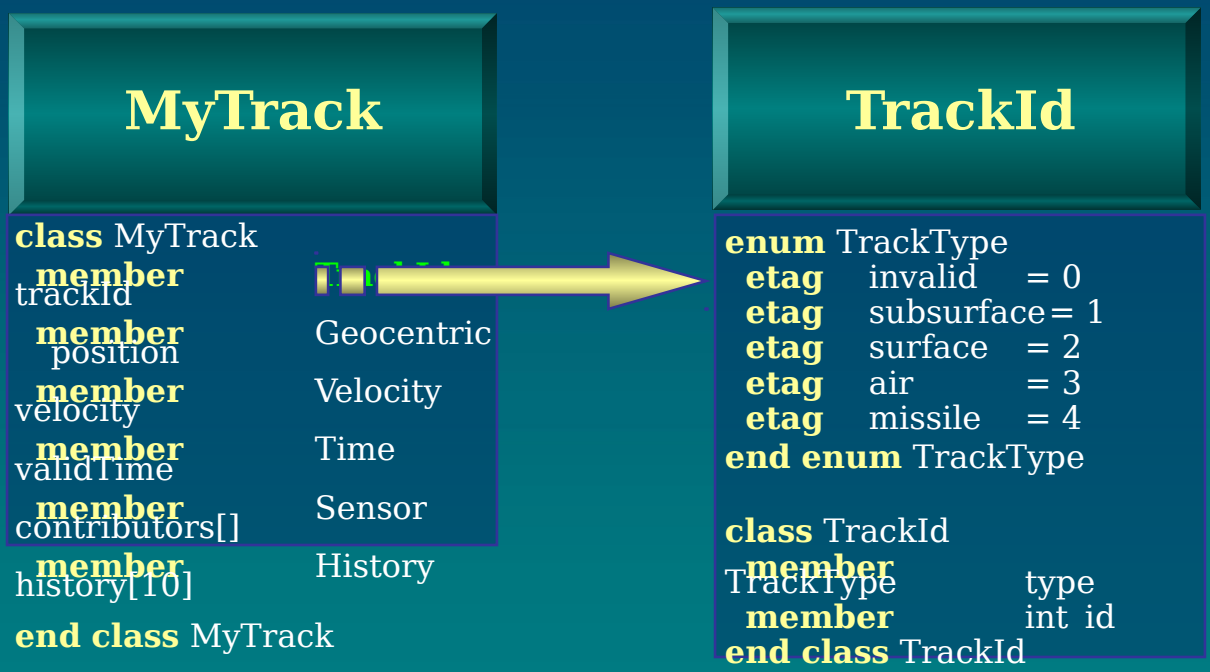
MyTrack

```
class MyTrack
  member TrackId trackId
  member Geocentric position
  member Velocity velocity
  member Time validTime
  member Sensor contributors[]
  member History history[10]
end class MyTrack
```

© 2001 Solipsys Corporation

Defining Classes Within Classes

Classes may be nested as shown here with TrackId.



© 2001 Solipsys Corporation

Using Pre-existing Classes

Classes from existing libraries may also be used to avoid reinventing the wheel.

MyTrack

```
class MyTrack
  member    TrackId     trackId
  member    Geocentric  position
  member    Velocity    velocity
  member    Time        validTime
  member    Sensor      contributors[]
  member    History     history[10]
end class MyTrack
```

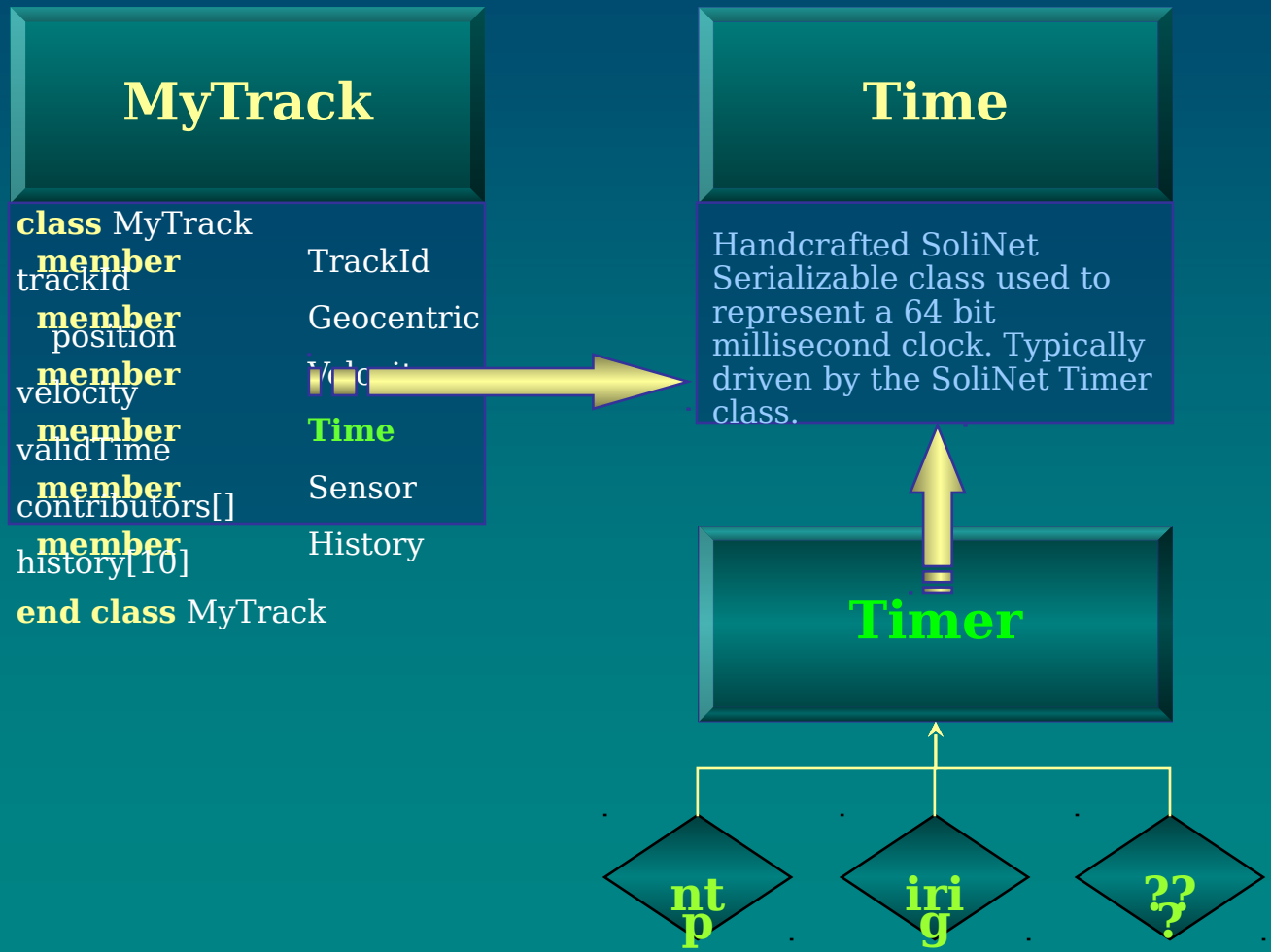
Geocentric

```
// SoliMath X,Y,Z Position
// Earth Centered / Earth Fixed
//
class Geocentric
  member    double      x
  member    double      y
  member    double      z
end class TrackId
```

© 2001 Solipsys Corporation

Utilizing SoliNet Time

The SoliNet time class can be used to provide a simple and consistent notion of system, or scenario time.



© 2001 Solipsys Corporation

Collections: The List

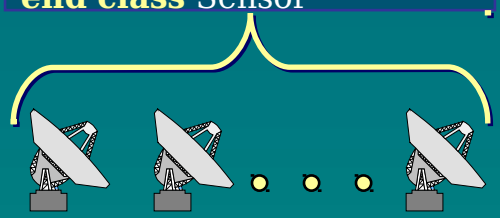
Object collections, in this case an unbounded list, are supported by the meta language.

MyTrack

```
class MyTrack
  member TrackId trackId
  member Geocentric position
  member Velocity velocity
  member Time validTime
  member Sensor contributors[]
  member History history[10]
end class MyTrack
```

Sensor

```
// The [] notation represents a
// collection of 0..n objects.
//
class Sensor
  member Geocentric position
  member double band
  member double power
end class Sensor
```



© 2001 Solipsys Corporation

Collections: The Vector

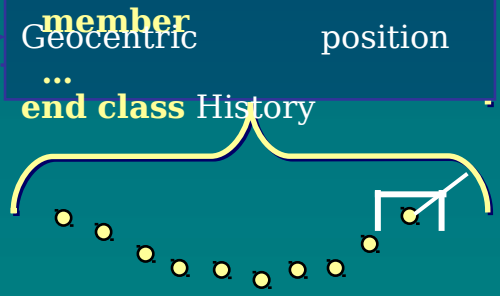
Collections may also have hard boundaries placed upon them.

MyTrack

```
class MyTrack
  member    TrackId     trackId
  member    Geocentric  position
  member    Velocity    velocity
  member    Time        validTime
  member    Sensor      contributors[]
  member    History     history[10]
end class MyTrack
```

History

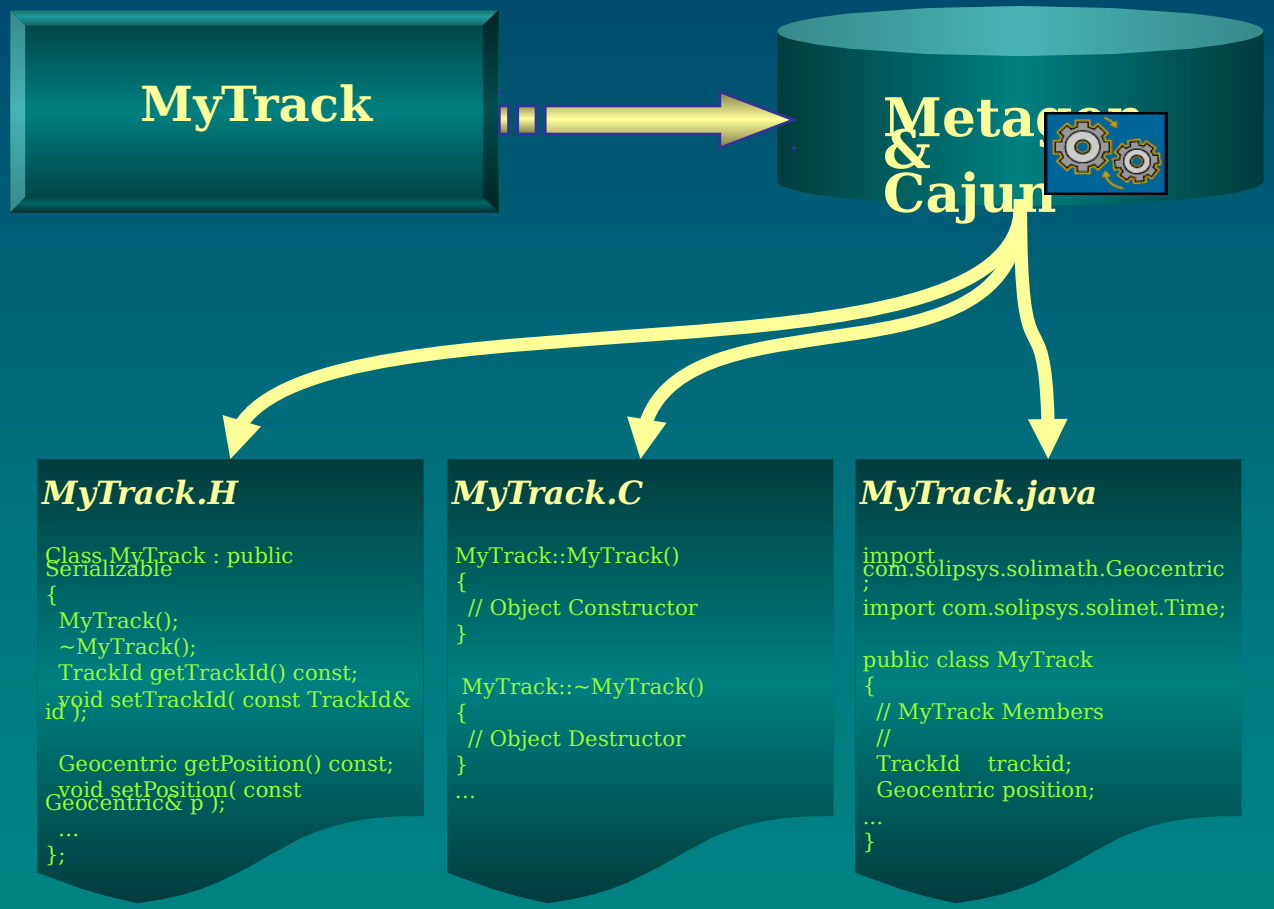
```
// The [10] notation represents a
// collection of 10 objects.
//
class History
  member    TrackId     trackId
  member    Geocentric  position
  ...
end class History
```



© 2001 Solipsys Corporation

Code Generation Phase

Feeding the meta source into the compiler will produce all of the C++ and Java code needed to freely exchange the objects.

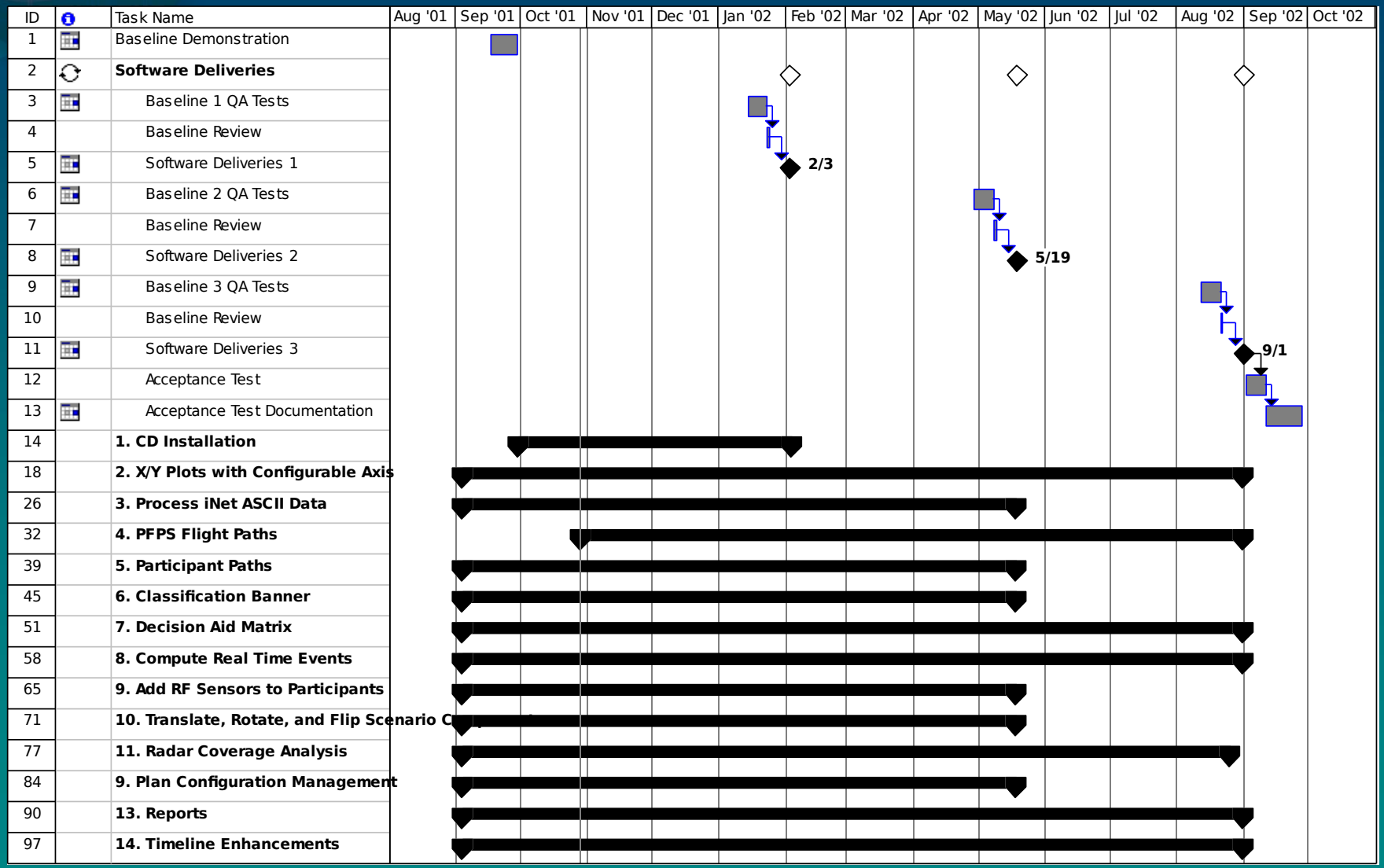


© 2001 Solipsys Corporation

Software Development Plan

© 2001 Solipsys Corporation

Software Development Plan



© 2001 Solipsys Corporation

Software Development Plan

- **MS Project Plan**
  - **Software process**
  - **Build schedule**
  - **QA**
  - **Delivery schedule**
  - **Documentation**
  - **Acceptance test**

© 2001 Solipsys Corporation

New Requirements

- Installation CD
- Real Time Analysis Plots
- Process ASCII Formatted Trajectory Data
- Personal Flight Planning System (PFPS) Compatibility
- Vehicle Waypoint Trajectories - Special Cases
- Classification Banner
- Decision Aid Matrix
- Compute Real Time Events
- Mobile Sensors
- Scenario Plan Manipulation
- Radar Coverage Analysis
- Plan Configuration Management
- Plan Reports
- Timeline Enhancements

© 2001 Solipsys Corporation

Installation CD

© 2001 Solipsys Corporation

Installation CD

- Currently using InstallShield, Java edition
  - One installer for Win32, Solaris, etc.
  - Java based, requires a JVM pre-installed
  - Some minor issues, may be resolved with future InstallShield releases?
- Other options
  - Self-extracting zip file
    - Simplest option
    - Less flashy
    - Won't do system dependent tasks
  - Multiple installers for multiple systems
    - InstallShield/Wise for Win32
    - Solaris packages
    - Other platforms, e.g., Linux rpm
    - If automated, is this really more effort?

© 2001 Solipsys Corporation

Real Time Analysis Plots

© 2001 Solipsys Corporation

Real-Time Data Presentation

- **Tactical Situation Display**
  - Situation awareness
  - Visual aids
- **Analysis Plots**
  - User specified during mission planning
  - Quantitative

© 2001 Solipsys Corporation

Plots

- iNet Kinematic Contents
  - Position, Velocity, Acceleration, Valid Time
  - Track State Covariance
  - Attitude
- iNet Attribute Contents
  - Track Quality
  - VID/SID
  - Security Level
- Derived Quantities
  - Down range, Cross range, Altitude
  - Aspect Angle
  - Relative spherical position
  - Closest Point of Approach (CPA)
  - Time-to-go
  - Residuals

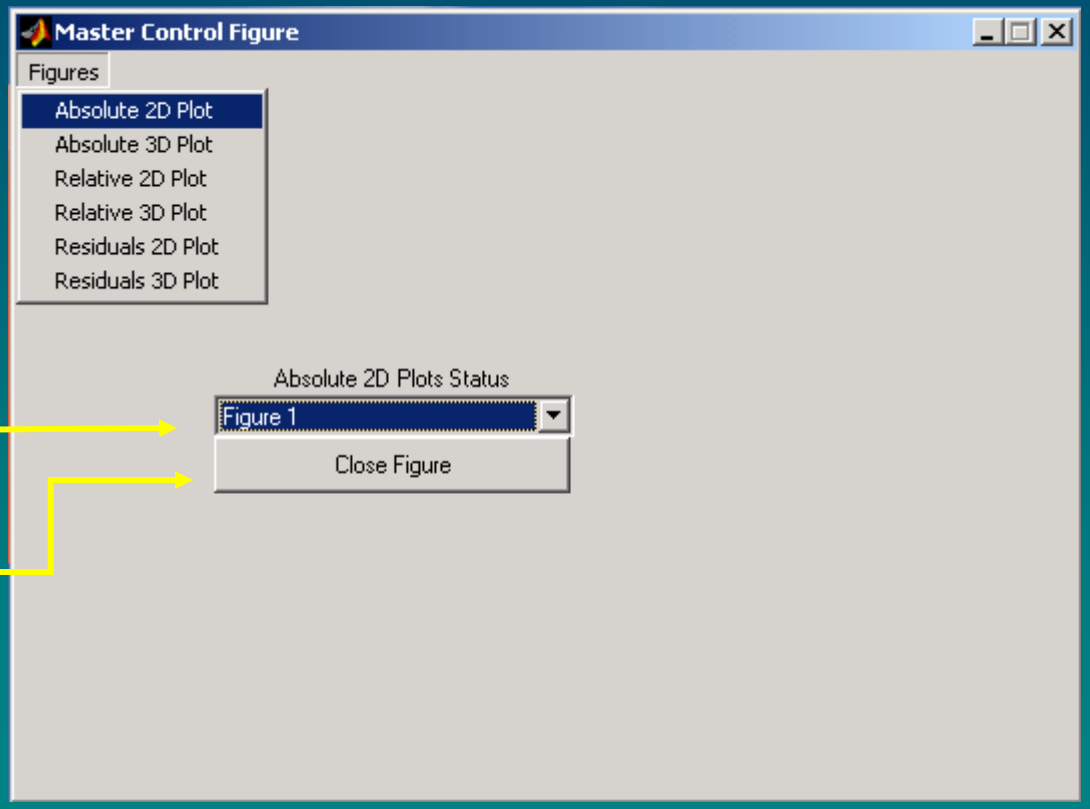
© 2001 Solipsys Corporation

Plot Figure Initiation

Add New Figures

List of Current Figures

Close Selected Figure



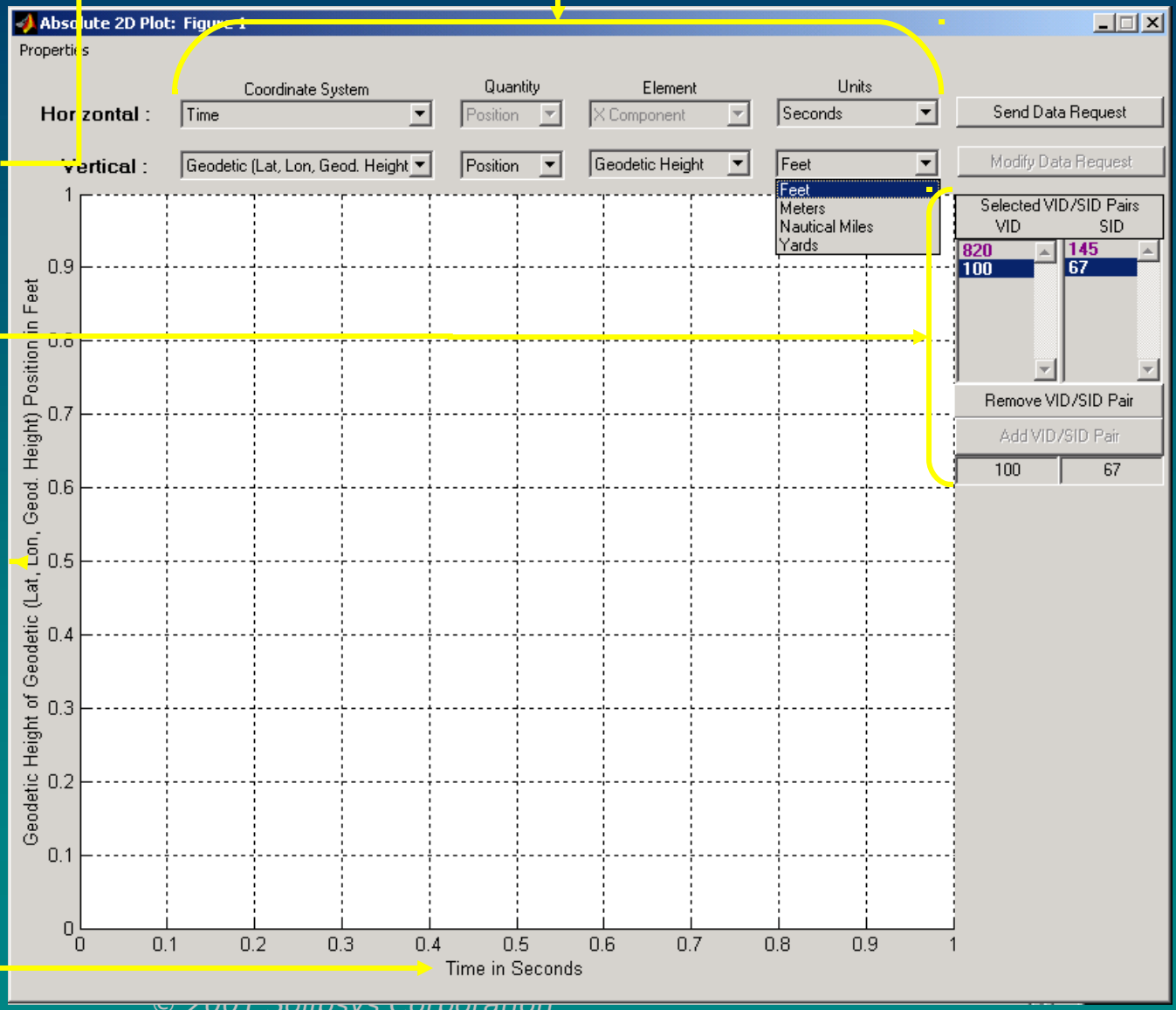
© 2001 Solipsys Corporation

Plot Definition

Adaptive Pop Menus

VID/SID Entry Panel

Adaptive Axis Labels



© 2001 Solipsys Corporation

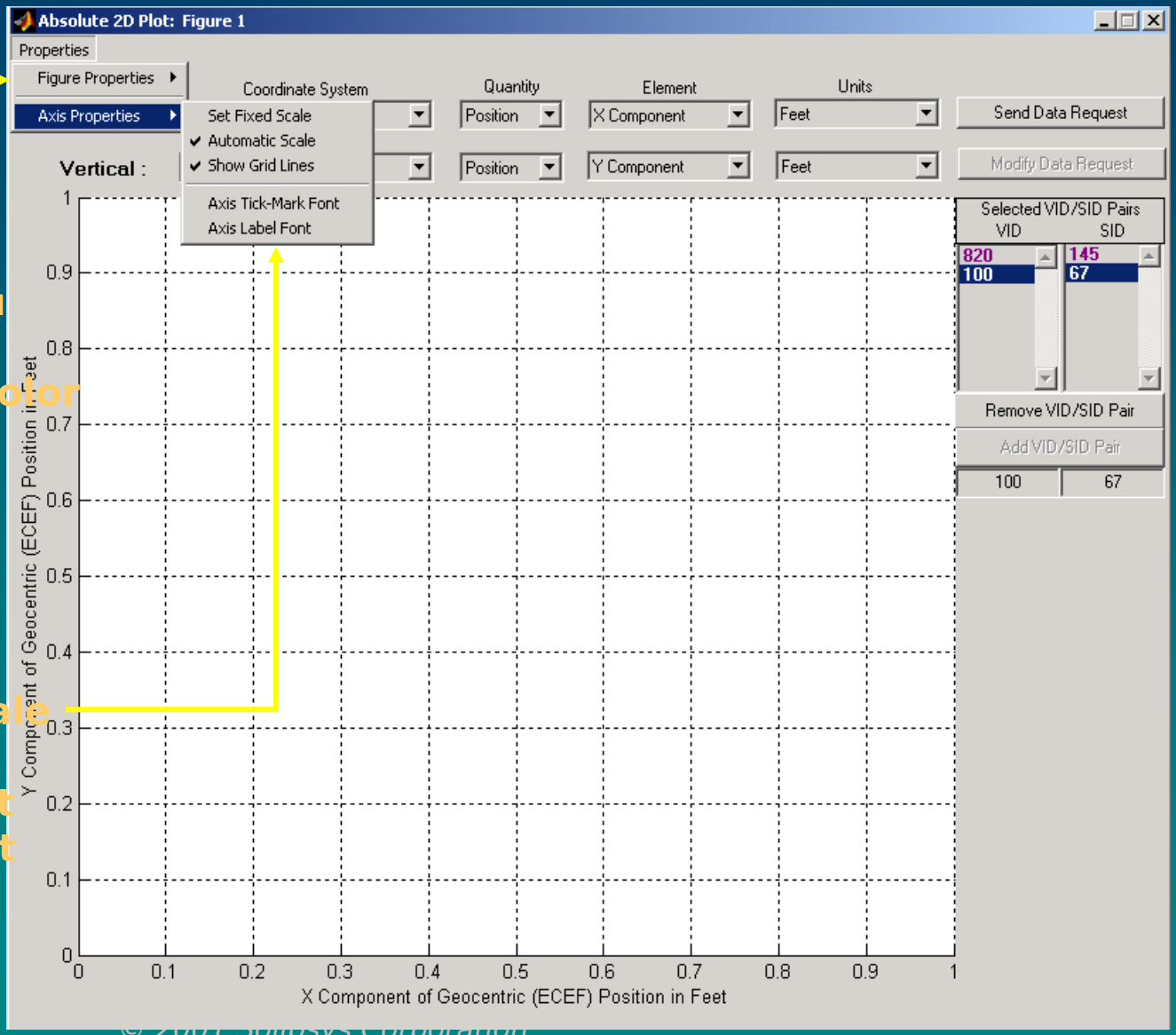
Plot Definition

Figure Properties Menu

- Modify background color
- Save figure settings

Axis Properties Menu

- Set fixed axis scales
- Turn on automatic scale
- Toggle axis grid lines
- Change tick mark font
- Change axis label font

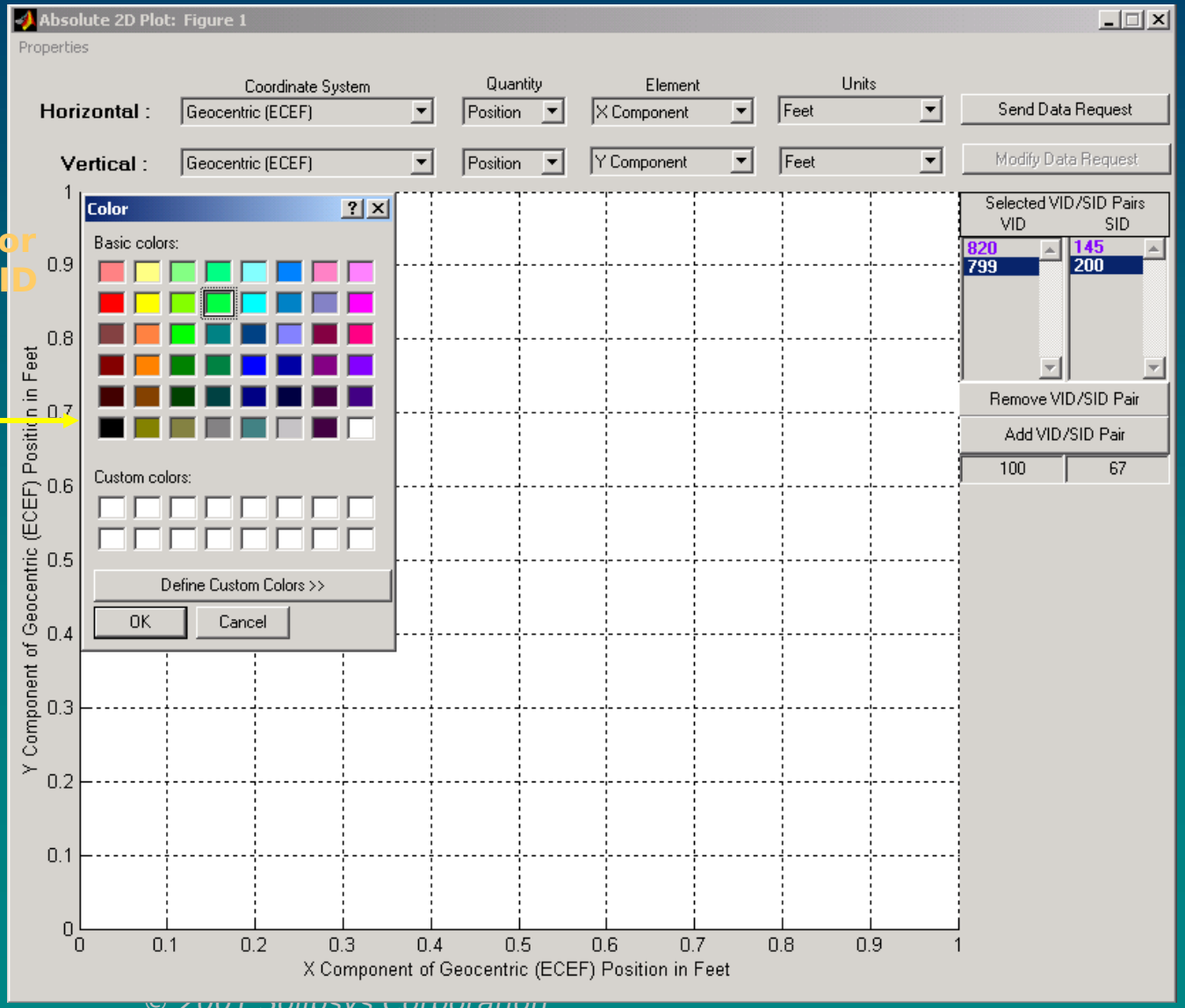


© 2001 Solipsys Corporation

Plot Definition

Custom Color Panel

- Modify background color
- Assign color to a VID/SID



© 2001 Solipsys Corporation

Real Time Analysis Plots
Build Schedule

- **Build 1 – 2/3/02**
  - X/Y plots with fixed scales, units, and increments for all explicit and implicit track data
- **Build 2 – 5/19/02**
  - Refinements
  - X/Y plots with fixed scales, annotations, and increments for additional plots that are reference point dependent
- **Build 3 – 9/1/02**
  - Refinements

© 2001 Solipsys Corporation

Process ASCII Formatted Trajectory Data

© 2001 Solipsys Corporation

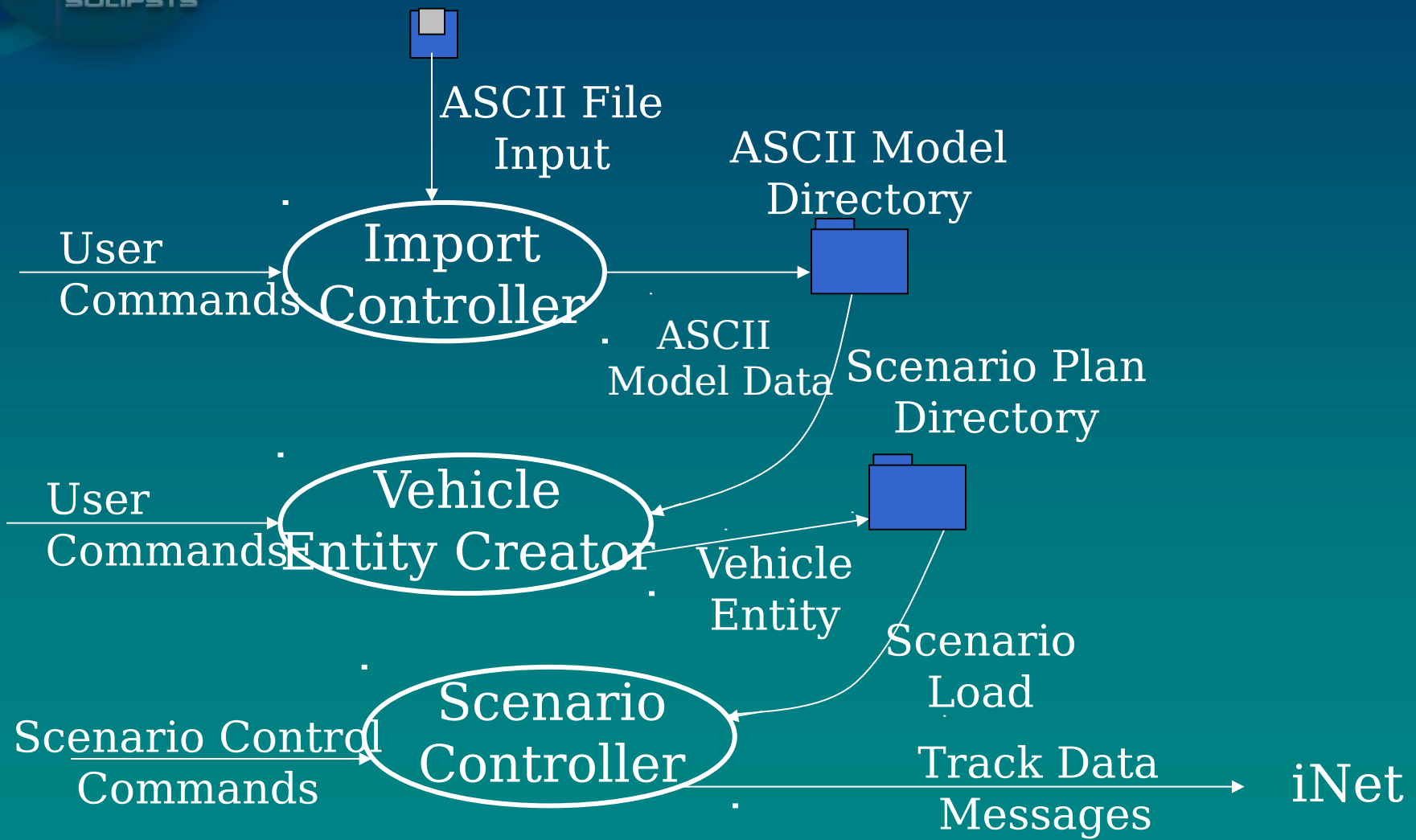
Process ASCII Formatted Trajectory Data

Description

- **Mission Tool shall import ASCII formatted trajectory data**
  - **Time, position, rates, optional orientation data**
- **ASCII model imported into any scenario as a vehicle entity**
- **Scenario execution sends requisite planned position and Track Data messages**

© 2001 Solipsys Corporation

Process ASCII Formatted Trajectory Data



© 2001 Solipsys Corporation

Process ASCII Formatted Trajectory Data

Build Schedule

- **Build 1 – 2/3/02 Full Capability**
- **Build 2 – 5/19/02 Refinements**

© 2001 Solipsys Corporation

Personal Flight Planning System (PFPS) File Compatibility

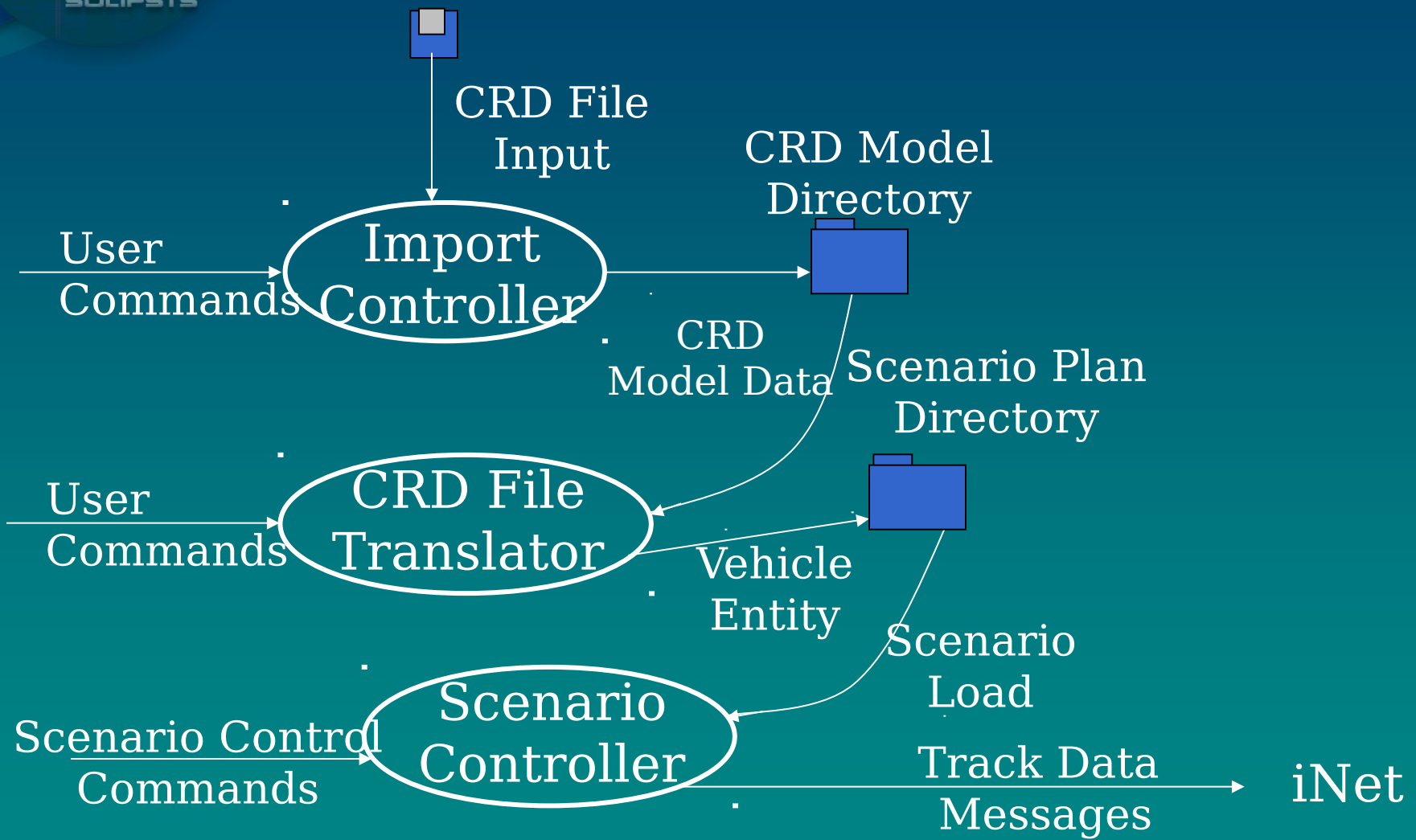
© 2001 Solipsys Corporation

PFPS File Interchange

- Recently received requisite Common Route Definition (CRD) Format - from TAMPS IDD
- Requirement is to translate CRD format to Waypoint based format
  - Full compatibility with current Mission Tool for route display, rehearsal, and real time mission monitor.

© 2001 Solipsys Corporation

Common Route Definition File to Waypoint Trajectory Definition



© 2001 Solipsys Corporation

Vehicle Waypoint Trajectories Special Cases

© 2001 Solipsys Corporation

Participant Locations as Fixed Points

- Many participants are positioned at a specific location throughout a scenario
  - Aux Sensors
    - MATSS
  - Interceptor launch vessel
    - USS Lake Erie
- Others follow a simple racetrack to remain near a specific location
  - G1 aircraft
  - P3 aircraft
- New feature will allow these participants to be added easily

© 2001 Solipsys Corporation

Stationary Participants

- User selects a vehicle, assigns a VID, and places into scenario
- If vehicle has only one waypoint, and minimum speed is zero, user can right click on waypoint and designate as stationary
- By default, vehicle will exist throughout scenario
  - If desired, start/stop times may be entered

© 2001 Solipsys Corporation

Racetrack Participants

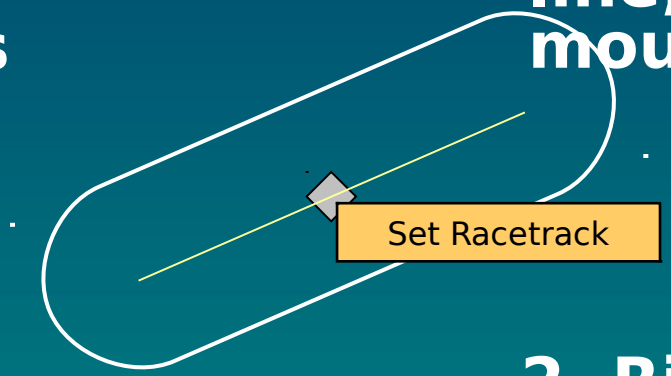
- User selects a vehicle, assigns a VID, and places into scenario
- If vehicle has only one waypoint, user can right click on waypoint and designate as racetrack
  - Axis line appears through waypoint to cursor
  - Mouse click sets axis line
  - Waypoint turn radius determines minimum axis length
- By default, vehicle will exist throughout scenario
  - If desired, start/stop times may be entered

© 2001 Solipsys Corporation

Racetrack Entry

1. User enters Waypoint

3. Drag out axis line, and click with mouse



2. Right-click popup menu item

4. Racetrack is drawn

© 2001 Solipsys Corporation

Build Schedule

- **Build 1 – 2/3/02**
  - **Full Capability**
- **Build 2 – 5/19/02**
  - **Refinements**

© 2001 Solipsys Corporation

Classification Banner

© 2001 Solipsys Corporation

Classification Banner Design

- **Description**
  - User shall be able to attach a classification level to any scenario
  - Classification level shall be displayed prominently on all displays and generated reports
  - Classification of a scenario shall be password-protected
- **Design Considerations/ Issues**
  - Classification banner on displays needs to reflect the higher of the current scenario classification and the track database classification as determined by the security level sent in any live track messages
  - Any frames that are printed via the GUI need to include classification level

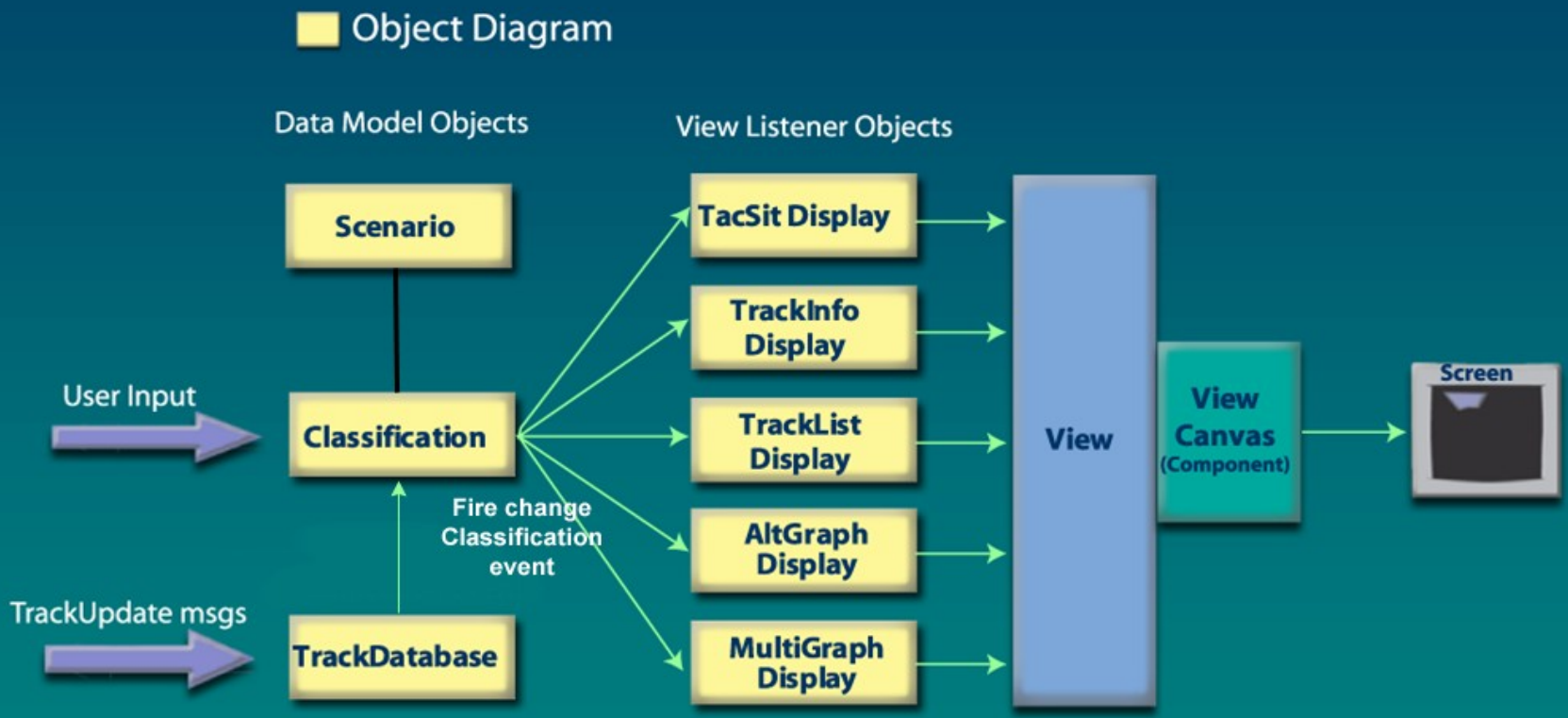
© 2001 Solipsys Corporation

Classification Banner Design

- **Data Flow**
  - **GUI will send GUIScenarioControl message to tell server to change scenario classification**
    - **Add new action SetScenarioClassification**
    - **Add classificationCode field**
    - **Add password field**
  - **Server will use SrvScenarioControlResponse message to send scenario classification to GUI**
    - **Add classificationCode field**
  - **Report Generator (on Server) will obtain classification level from the Scenario**

© 2001 Solipsys Corporation

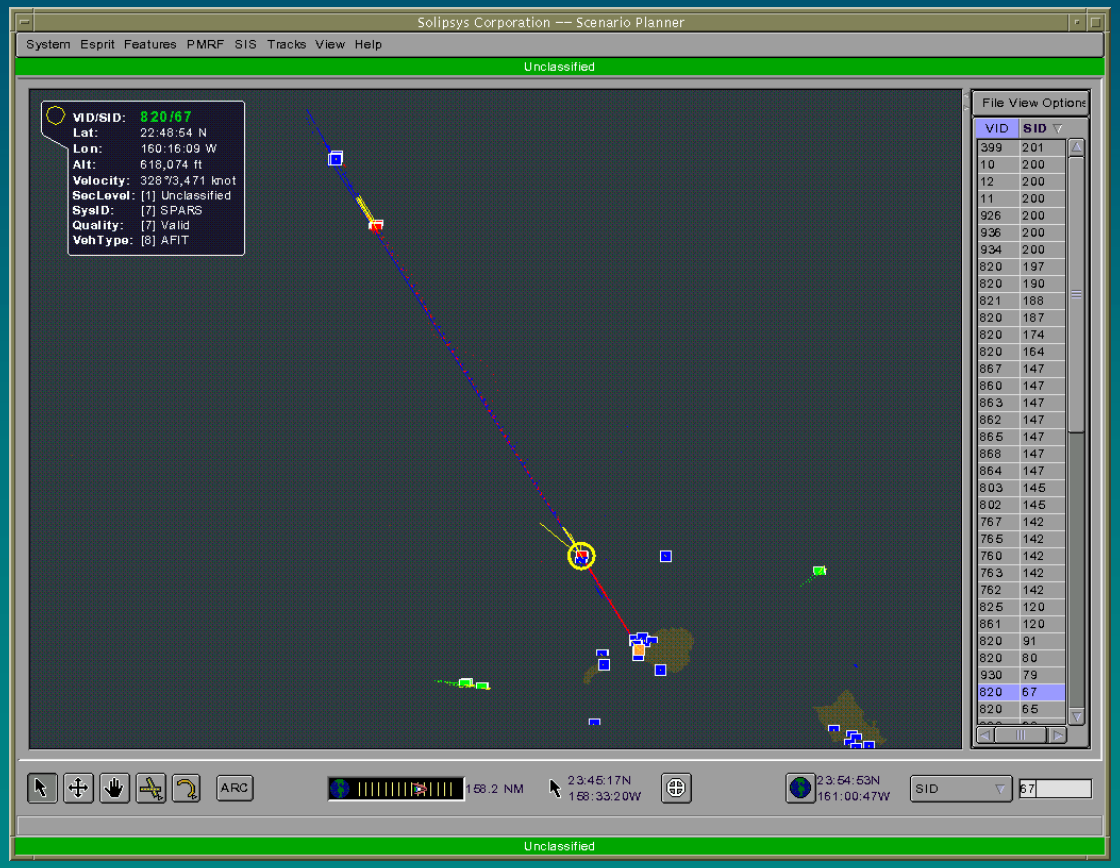
Classification Banner Design



© 2001 Solipsys Corporation

Classification Banner Design

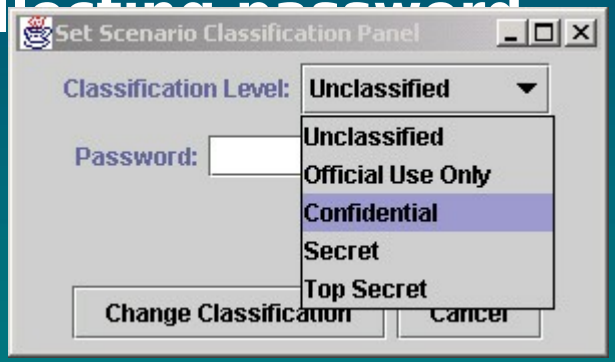
- **GUI**
  - **TacSit with Classification Banner**



© 2001 Solipsys Corporation

Classification Banner Design

- **GUI (cont'd)**
  - **New screen to be added for changing scenario classification level and collecting password**



© 2001 Solipsys Corporation

Classification Banner Design

- **GUI (cont'd)**
  - **Classification Banner to be added to the following existing screens**
    - **Track List**
    - **Track Information**
    - **Alt Graph**
    - **Multigraph**
  - **Classification Banner to be added to the following reports**
    - **Waypoint Report**
    - **Time Interval Report**
    - **Planned Position Report**
    - **Sensor Report**
    - **New reports**

      (Vehicle State Report, Vehicle Timeline Report, Scenario Timeline Report, Radar Assignment/ Coverage Report, Participant Location Map)

© 2001 Solipsys Corporation

Classification Banner Design

- **Build Schedule**
  - **Build 1 – 2/3/02**
    - **Display of Classification Banner as higher of scenario classification and track database classification complete**
    - **Password-protected setting of scenario classification complete**
  - **Build 2 – 5/19/02**
    - **Classification Banner added to existing reports and new reports included in this build**
    - **Classification Banner complete**

© 2001 Solipsys Corporation

Decision Aid Matrix

© 2001 Solipsys Corporation

Decision Aid

- **Description**
  - **The user shall be able to define and enter operational constraints which will be monitored either automatically or manually during rehearsals or the actual event.**
  - **Upon selection of a constraint type, the user shall be presented with a list of predefined parameters required to completely express the constraint.**
  - **During the event, the automatically monitored constraints shall be evaluated on a periodic basis, and the status of each shall be presented in the Constraint Matrix.**

© 2001 Solipsys Corporation

Decision Aid

- **Constraint Category**
  - Level of grouping for display of the Constraint Matrix
  - The user shall be prompted to enter a Category for each constraint.
  - Example Categories:
    - Aux Sensors
    - Pre-Launch(Target)
    - Pre-Launch (Intercept)
    - IP

© 2001 Solipsys Corporation

Decision Aid

- **Constraint Type**
  - Distance
  - Manual
  - In The Box
  - Sensor On Line
  - Sensor On Target
  - Off Nominal (Distance)
  - Off Nominal (Time)
  - CPA

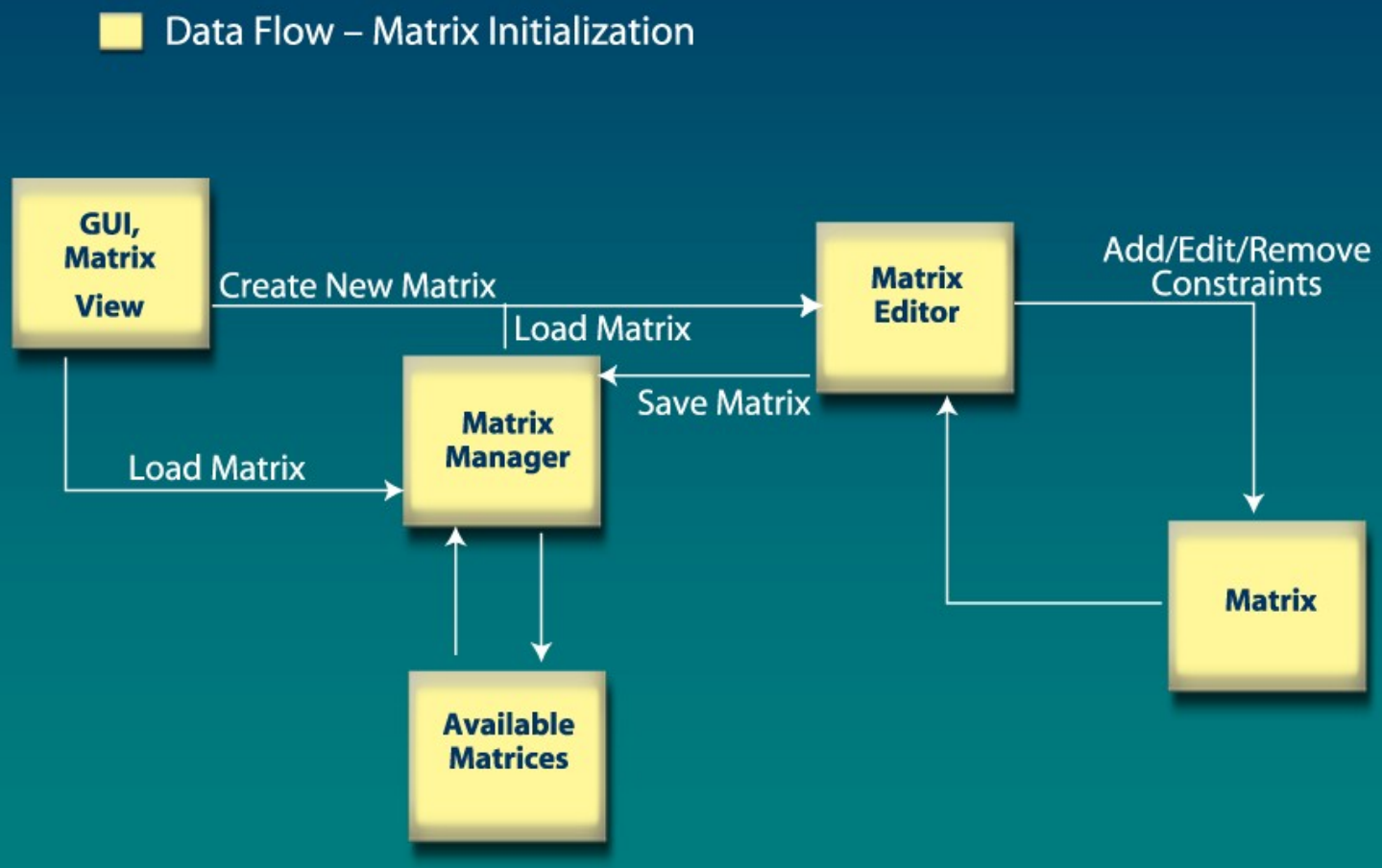
© 2001 Solipsys Corporation

Constraint Violation Presentation

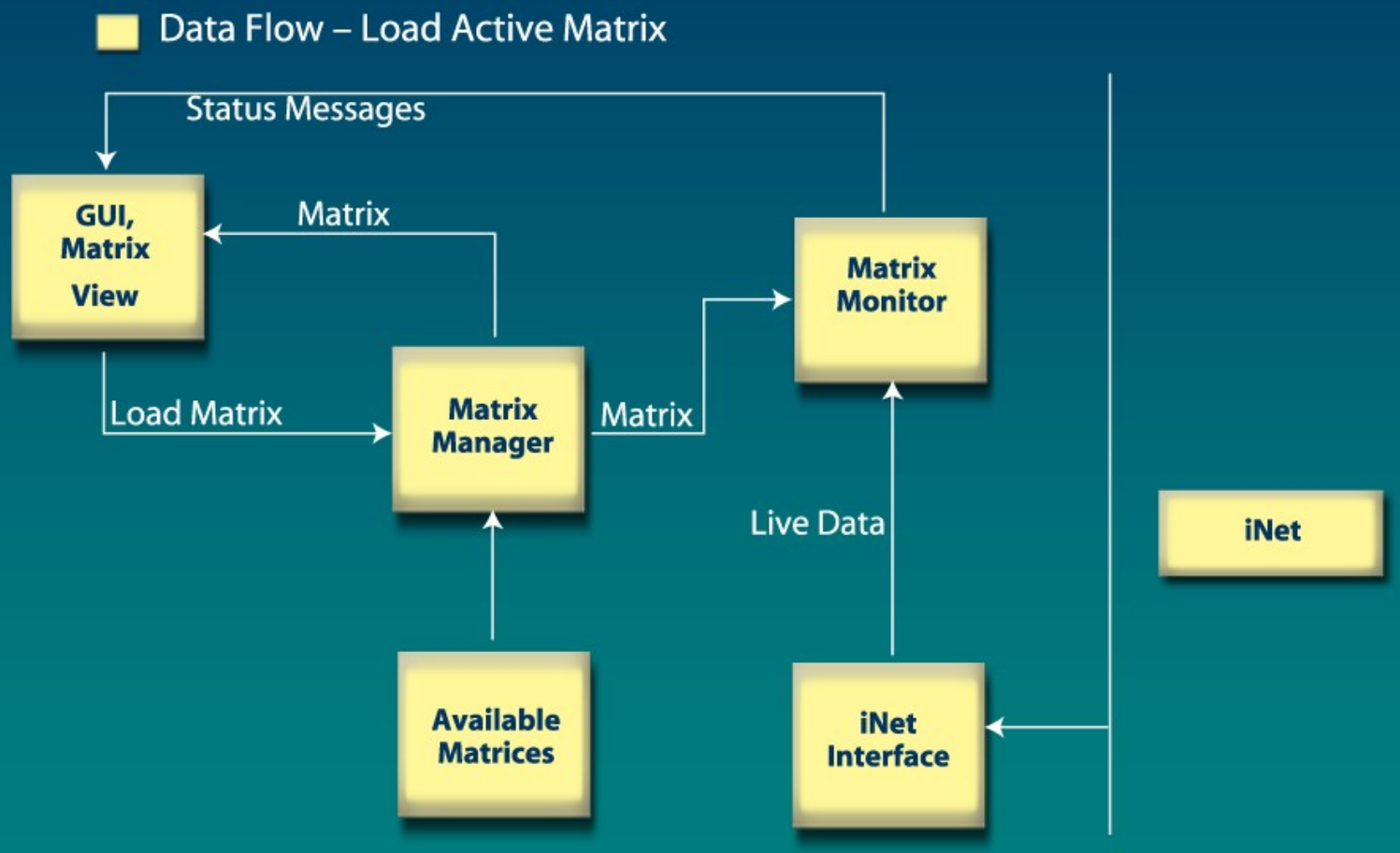
Type	Constraint Matrix	Symbology
Distance	Yes	Blinking Symbol
Manual	Yes	
In The Box	Yes	Blinking Symbol
Sensor Online	Yes	Icon 
Sensor On Target	Yes	Icon 
Off Nominal (Distance)	Yes	Blinking Symbol
Off Nominal (Time)	Yes	Blinking Symbol
CPA	Yes	Blinking Symbol

© 2001 Solipsys Corporation

Decision Aid

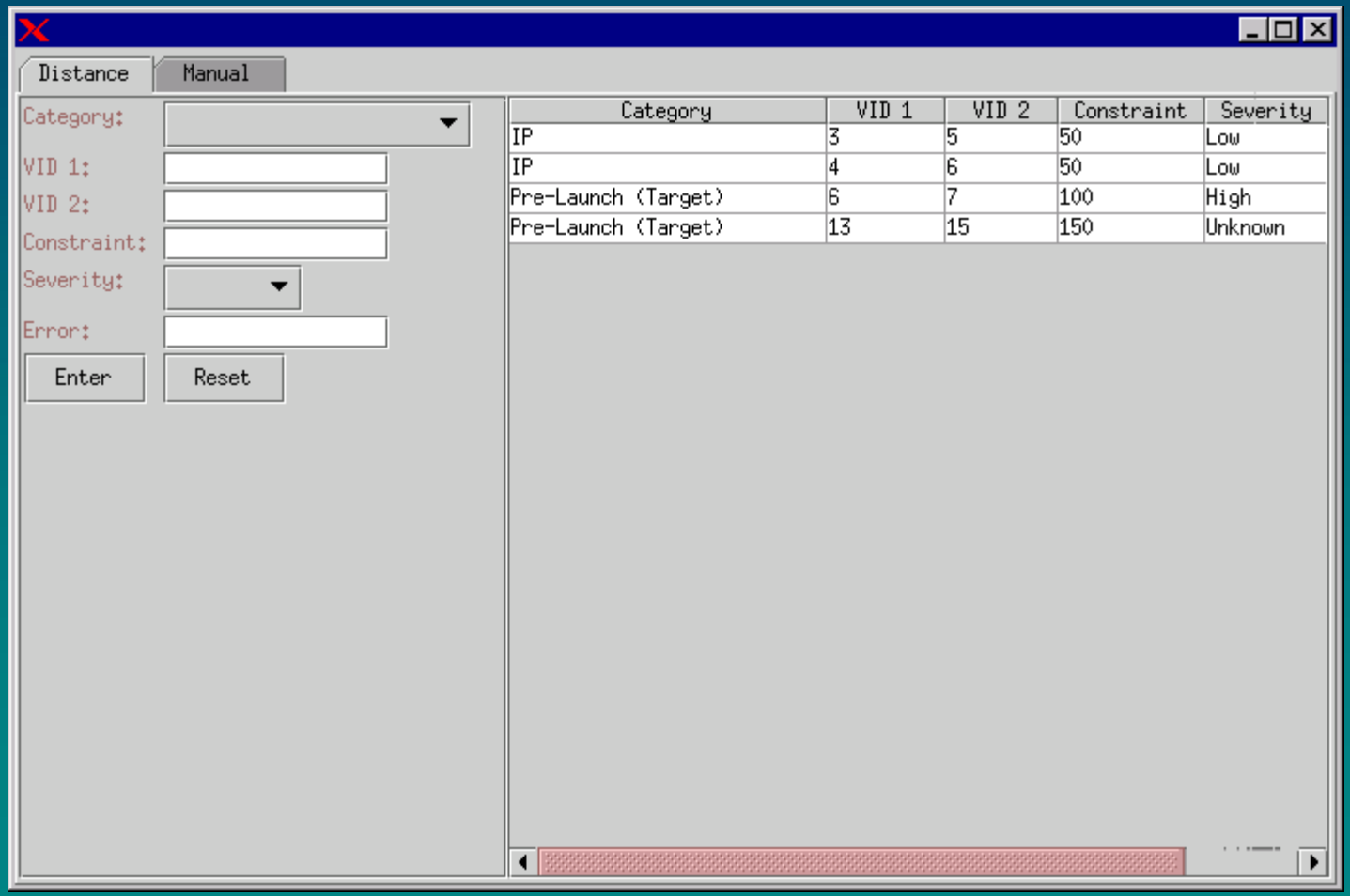


Decision Aid



Decision Aid

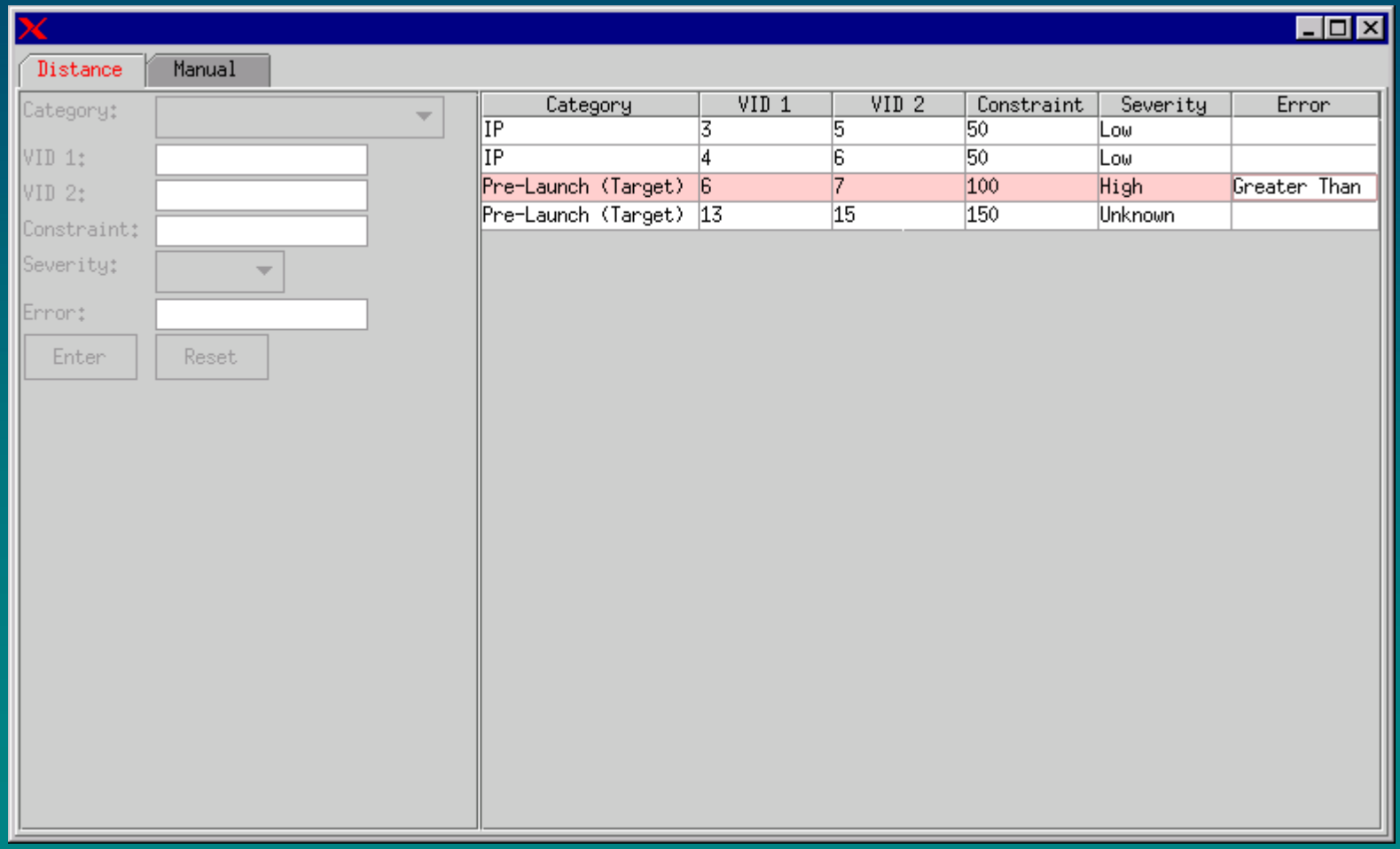
- **GUI – Constraint Matrix Editor**



© 2001 Solipsys Corporation

Decision Aid

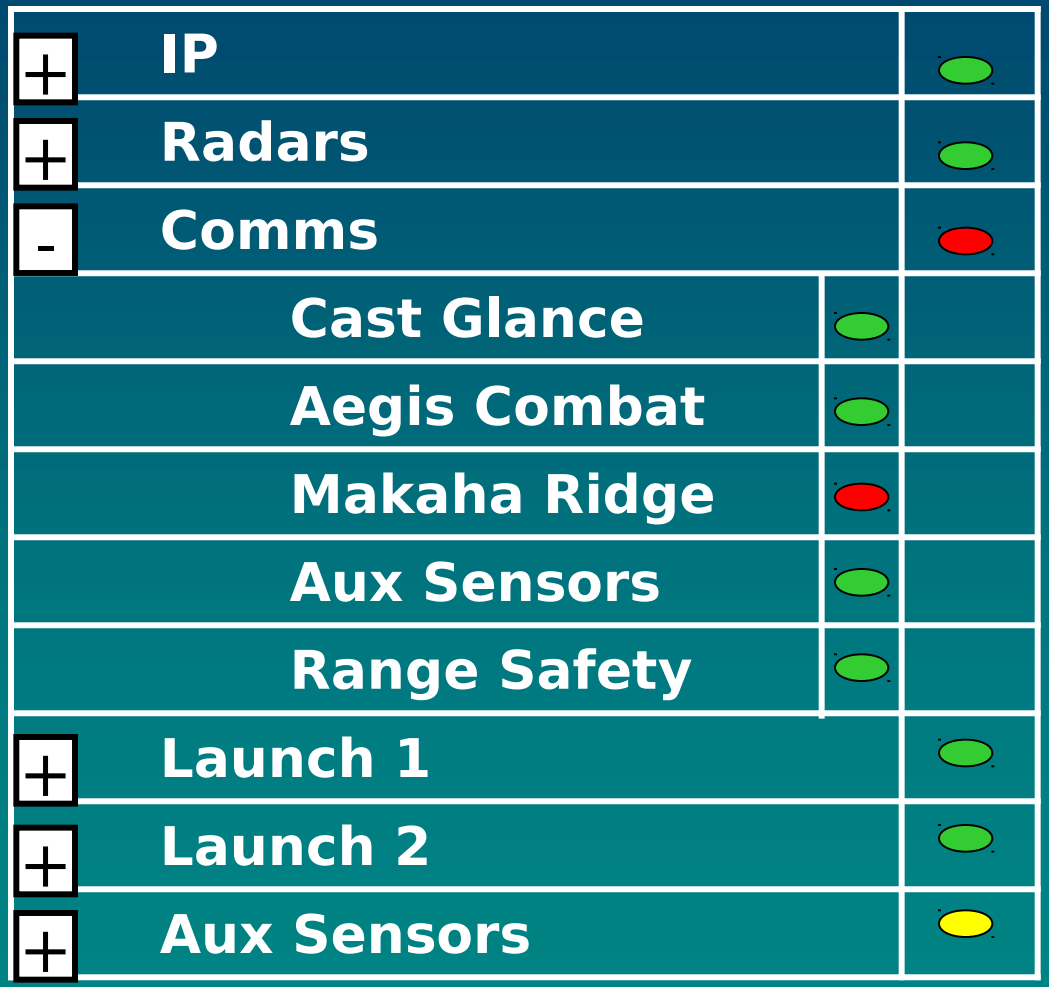
- **GUI – Constraint Matrix View**



© 2001 Solipsys Corporation

Decision Aid Matrix – Status Presentation

Decision Aid Matrix – Status Presentation



© 2001 Solipsys Corporation

Decision Aid

- **Build Schedule**
  - **Build 1 – 2/3/02**
    - **Constraint Entry Panel Complete**
    - **Constraint definition messages fully defined and implemented**
    - **Constraint monitoring algorithms defined and implementation completely defined**
    - **Minimal constraint monitoring implemented**
  - **Build 2 – 5/19/02**
    - **Constraint monitoring complete**
    - **Decision Aids complete**
  - **Build 3 – 9/1/02**
    - **Refinements**

© 2001 Solipsys Corporation

Compute Real Time Events

© 2001 Solipsys Corporation

Event Calculation & Display

- **Real-time event calculations that have instantaneous numerical outputs are:**
  - **Launch Time**
  - **Apogee Altitude**
  - **Apogee Time**
  - **IIP Altitude**
  - **IIP Time**
  - **Splash Time**
  - **Closest Point of Approach (CPA)**
  - **CPA Time OR Time-to-go**

© 2001 Solipsys Corporation

Event Calculation & Display

Instantaneous Real-time Events

- These calculations are automatically performed on the appropriate vehicles (such as TBM's) and may be performed on other vehicles as specified by the TDF interface
- The results will be displayed in several ways:
  - Columns in the track list
  - Track Tags
  - Track Details Box

© 2001 Solipsys Corporation

Event Calculation & Display

- **Launch Time Calculation**
  - Radar measurements are processed from SPARS
  - Time of first statistically significant motion is estimated using a multiple hypothesis batch estimator
- **Instantaneous Impact Prediction**
  - A gravitational ballistic prediction is performed using a variable step-size numerical integrator
  - Used to determine apogee, apogee time, altitude, altitude time, and splash
  - Used to determine CPA for ballistic objects
- **CPA Calculation**
  - Air Targets – straight line motion is assumed for CPA calculation
  - Both ground range CPA and slant range CPA will be computed and displayed

© 2001 Solipsys Corporation

Event Calculation & Display
Build Schedule

- **Build 1 – 2/3/02**
  - **Presentation designed and implemented**
  - **Event calculations designed**
  - **Messages defined**
- **Build 2 – 5/19/02**
  - **Implementation complete**
- **Build 3 – 9/1/02**
  - **Refinements**

© 2001 Solipsys Corporation

Mobile Sensors

Combined with Radar Coverage Analysis

© 2001 Solipsys Corporation

Scenario Plan Manipulation

© 2001 Solipsys Corporation

Mission Planning

- **Rotation**
  - User inputs origin and magnitude of rotation through GUI
  - All associated points are rotated about the the axis normal to the origin's local tangent plane
  - The rotation moves waypoints along a geodesic:
    - Orthogonal to the line of sight from the origin
    - A distance consistent with the specified change in azimuth
- **NOTE: Large rotations and translation can cause waypoint distortion – User can elect to preserve ground range separation and altitude of waypoints**

© 2001 Solipsys Corporation

Mission Planning

- **Translation**
  - **User can drag and drop OR manually enter north-east offset**
  - **All points are translated along a geodesic path by:**
    - **Moving the points in the direction indicated by the offset**
    - **Moving the points a distance indicated by the offset magnitude**
- **NOTE: Large rotations and translation can cause waypoint distortion – User can elect to preserve ground range separation and altitude of waypoints**

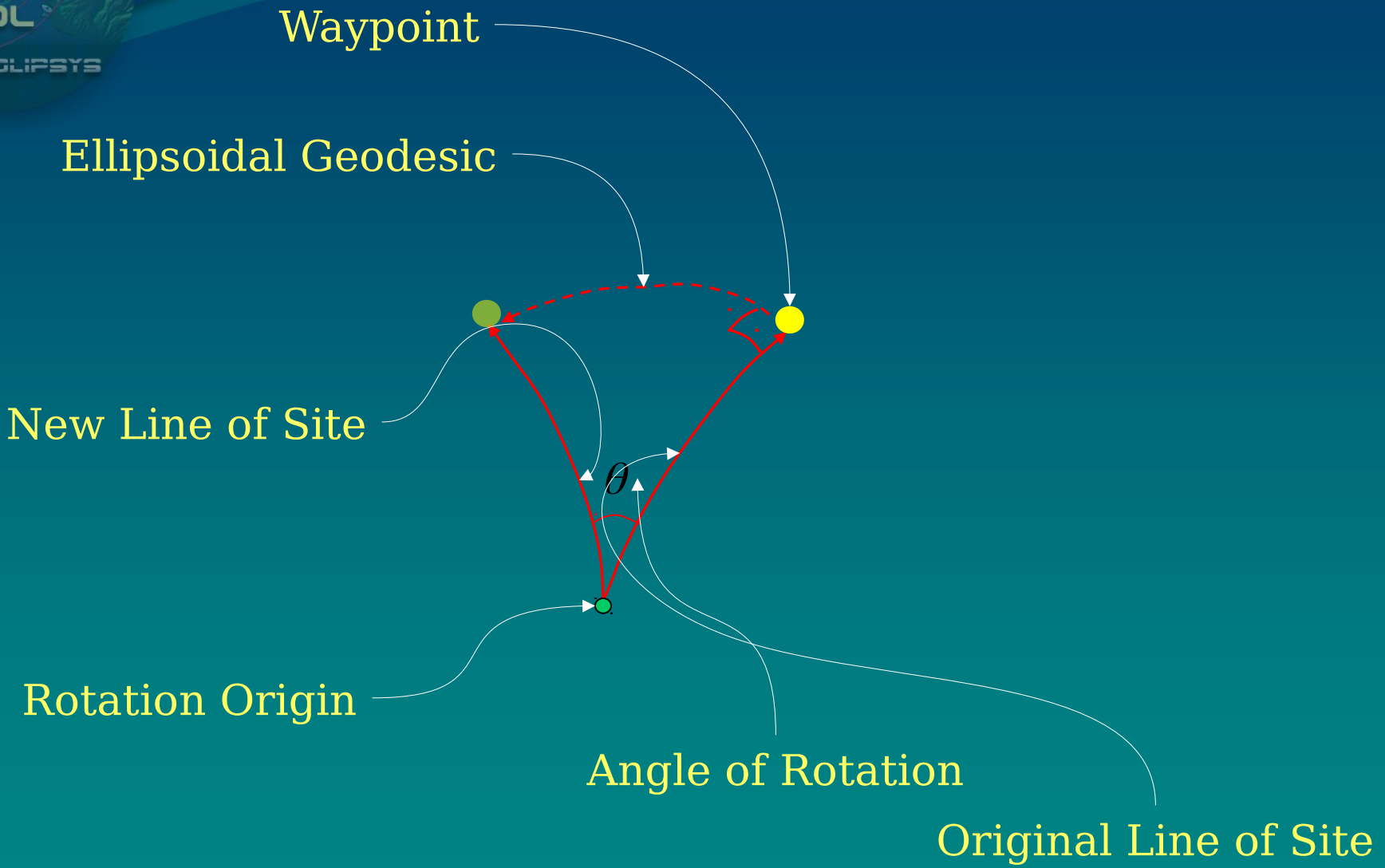
© 2001 Solipsys Corporation

Mission Planning

- **Mirror Flip**
  - **User specifies an origin and azimuth OR two geodetic points**
  - **Waypoints are “rotated” to the other side of this tangent line**

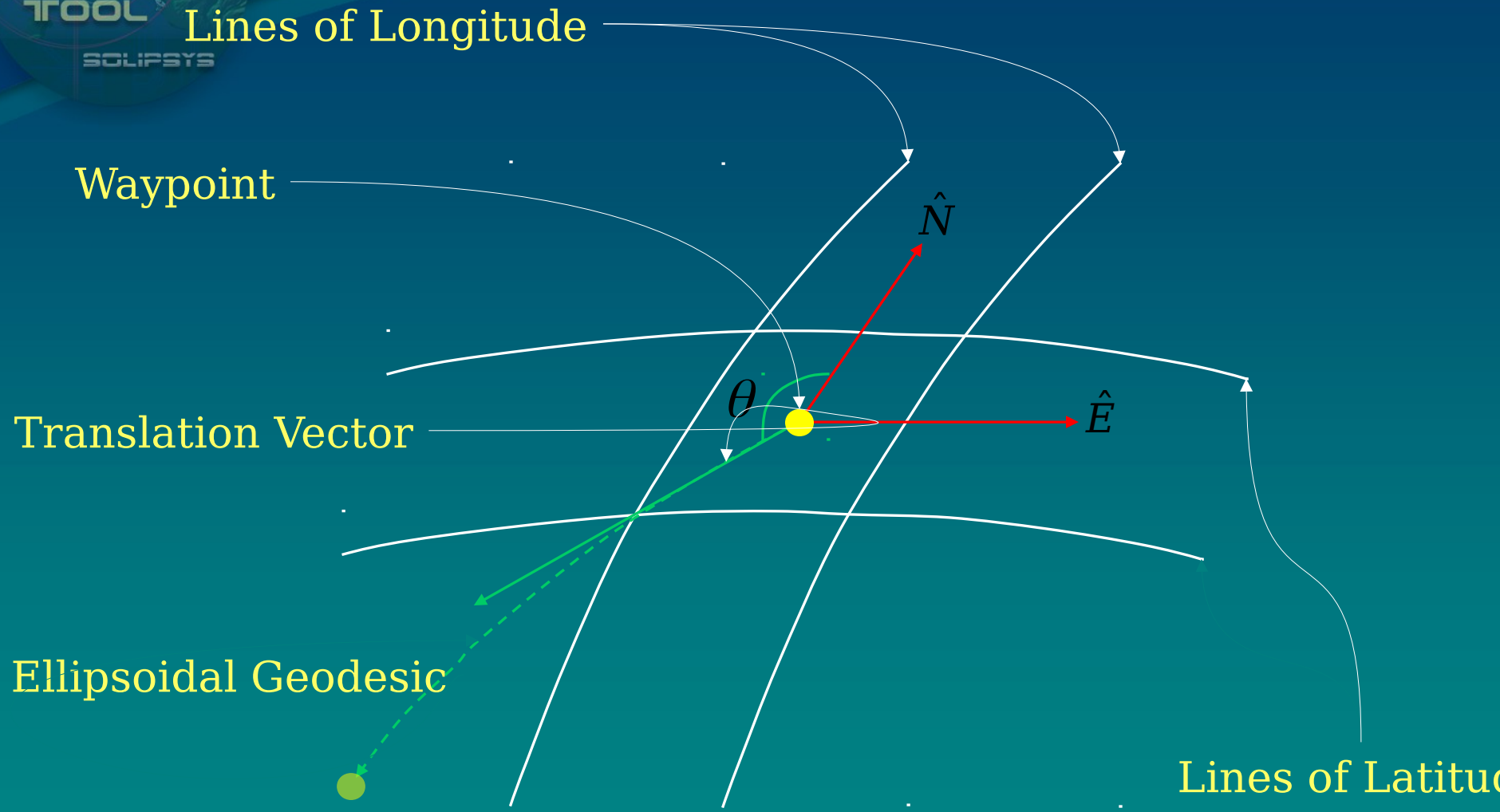
© 2001 Solipsys Corporation

Mission Planning (Rotation)



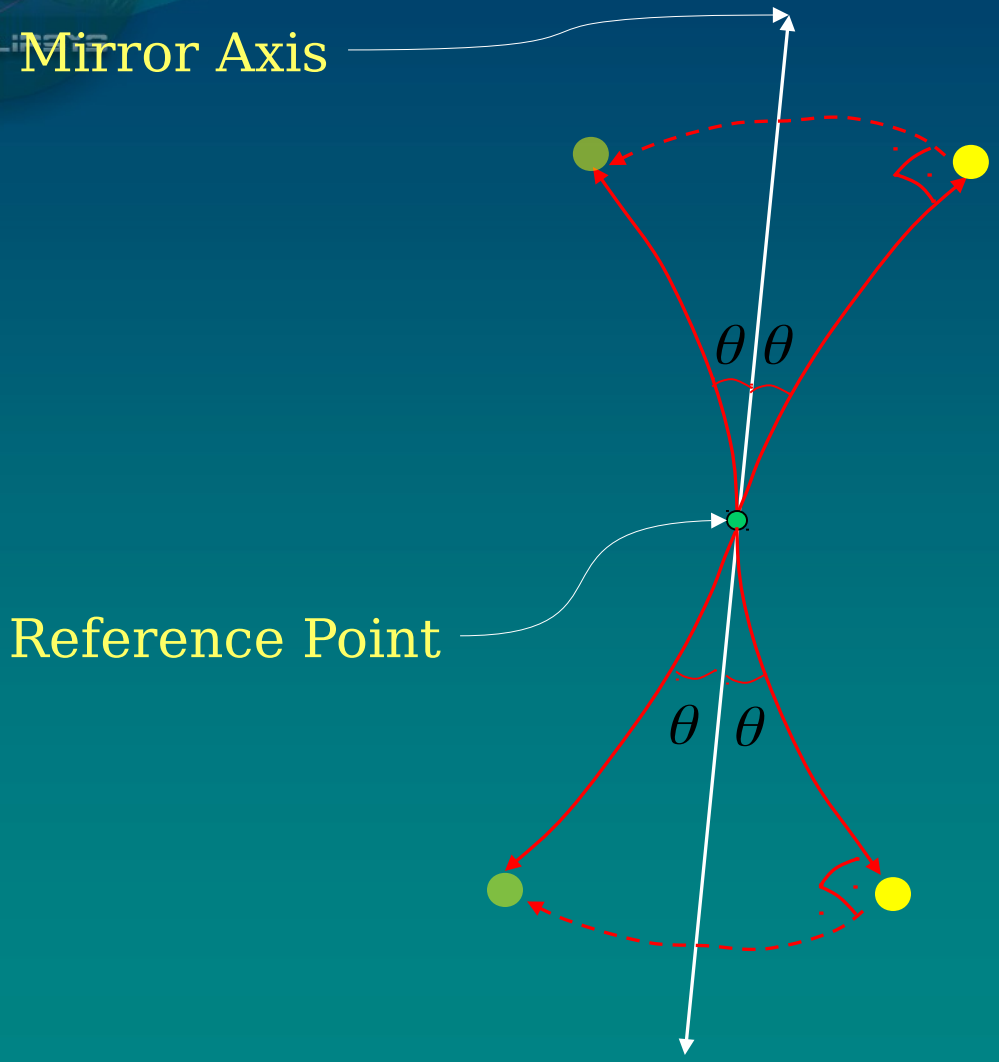
© 2001 Solipsys Corporation

Mission Planning (Translation)



© 2001 Solipsys Corporation

Mission Planning (Mirror Flip)



© 2001 Solipsys Corporation

Radar Coverage Analysis

© 2001 Solipsys Corporation

Modes of Range Mission Tool – Mission Planning

- **Mission Planning:**
  - **Sensor models are defined**
    - **Mode parameters**
  - **Sensor entities are defined**
    - **Stationary**
    - **Mobile**
    - **Shadow masks**
    - **Cut-outs**
    - **Vehicle reporting responsibility**
  - **Vehicles are defined**
    - **Waypoints**
    - **Timeline**
    - **RCS**
  - **Vehicles coverage analysis is performed**
    - **S/N analysis**
    - **Coverage summary**

© 2001 Solipsys Corporation

Modes of Range Mission Tool – Mission Preview

- **Mission Preview**
  - Graphical illustration of line-of-sight is indicated for each radar-target pairing
  - User alerted if a target completely loses coverage
  - Vehicles icons are displayed along with planned trajectory

© 2001 Solipsys Corporation

Modes of Range Mission Tool- Mission Rehearsal

- **Mission Rehearsal**
  - **Msg25 and/or Msg26 data can be distributed on the iNet**
  - **Track icons appear when sensors are “on track”**
  - **iNet data is simulated using sensor parameters and real-time data processing**

© 2001 Solipsys Corporation

Mission Planning (Sensors)

- **Pointing Radars**
  - **Each sensor model has configuration parameters including, but are not limited to:**
    - **Maximum detection range**
    - **Angular slew rates**
    - **Transmitter Power**
    - **S/N Threshold**
    - **RF Loop Gain**
    - **Beamwidth**
  - **A sensor model is unique to a sensor type but independent of sensor location and vehicle assignment**
- **Fixed Radars**
  - **Angular slew rates are irrelevant**
  - **Beam width is replaced by half-angle**

© 2001 Solipsys Corporation

Mission Planning (Sensors)

- **Sensor Entities**
  - **Stationary**
    - **Shadow masks – due to land masses, buildings, etc..**
    - **Radiation hazard cut-outs – due to EMI considerations, etc…**
    - **Boresight (Fixed radar only)**
  - **Mobile**
    - **Shadow masks, Radiation hazard cut-outs, and Boresight data are relative to platform heading**
  - **Coverage zones are input using comma delimited ASCII text format**

© 2001 Solipsys Corporation

Radar Coverage Analysis – Mission Planning (Sensors)

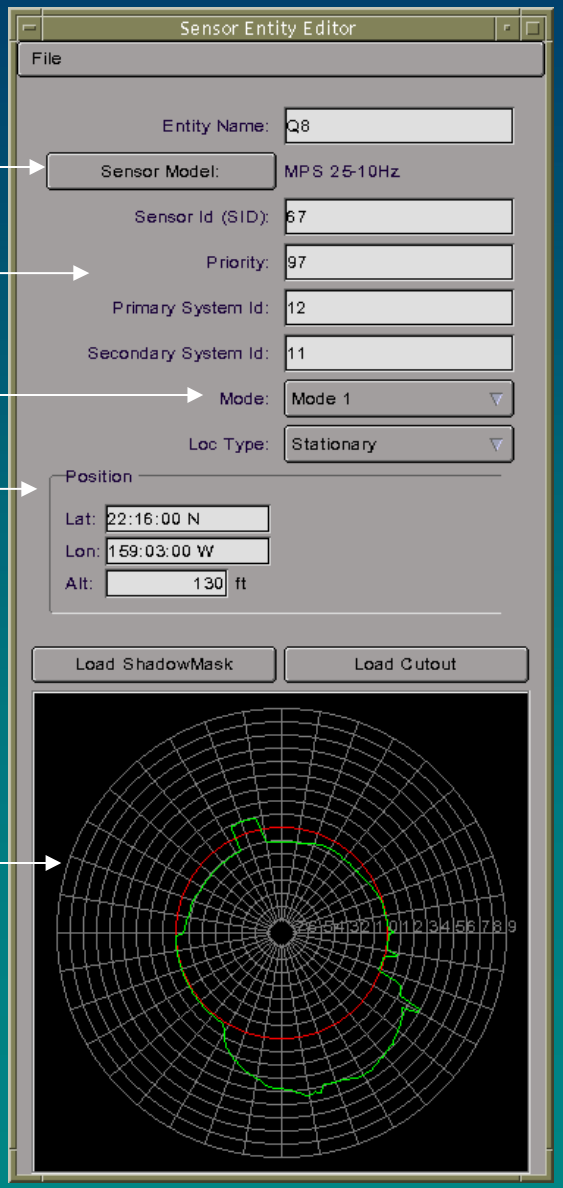
Load Sensor Specific Data

Enter Site Parameters

Select Sensor Operation Mode

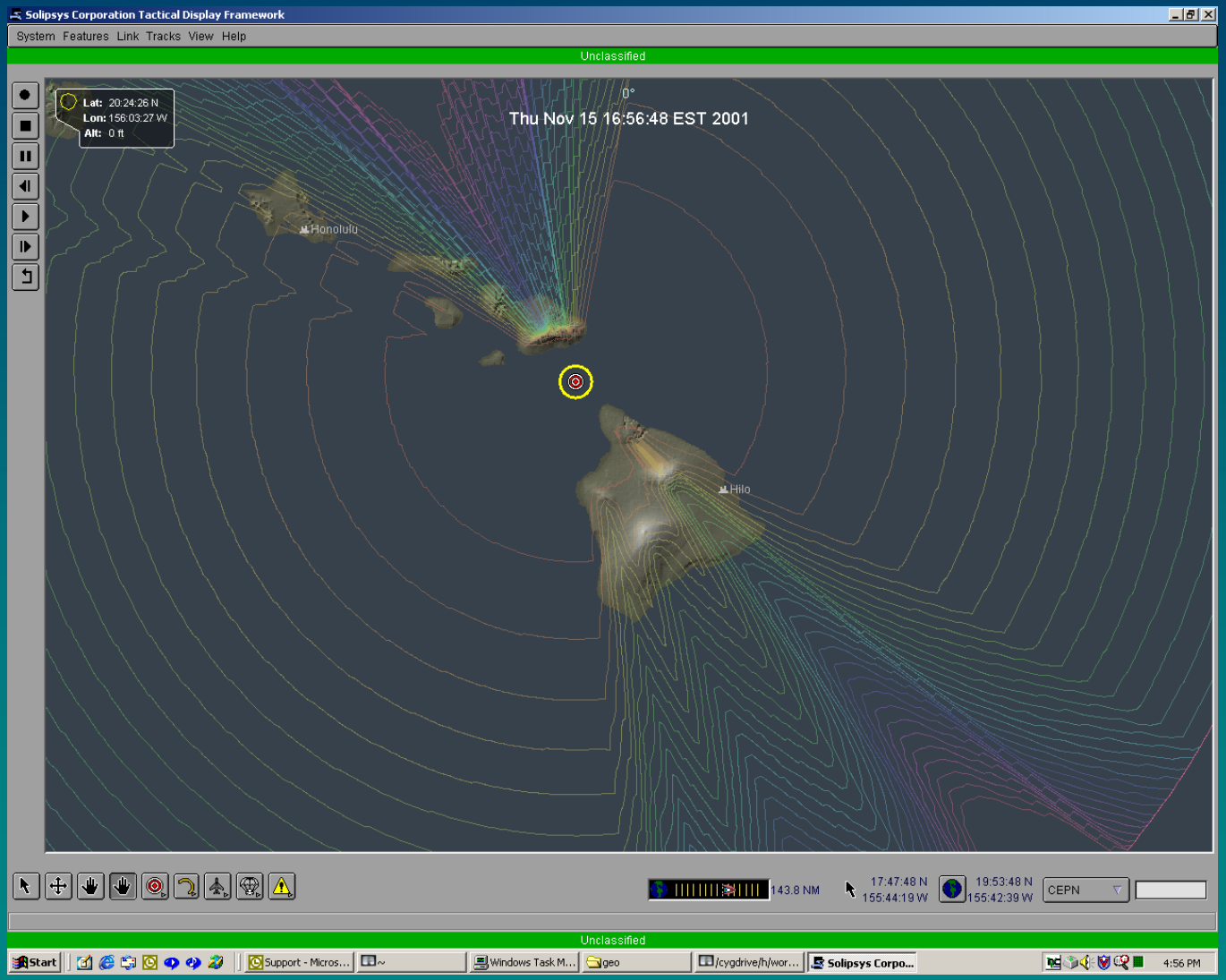
Stationary or Mobile

Shadow Mask and Cut-out Plot



© 2001 Solipsys Corporation

Mission Planning (Sensors)



© 2001 Solipsys Corporation

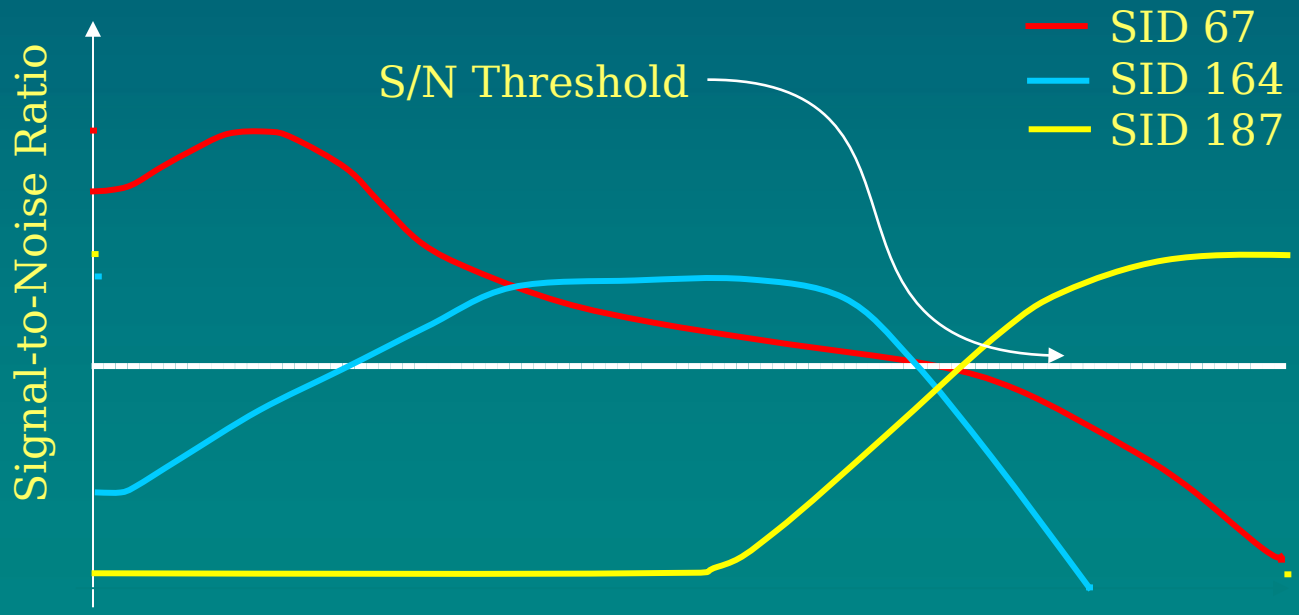
Mission Planning (Vehicles)

- Vehicle waypoints can be specified in the following ways:
  - Geodetic location and time-of-arrival
  - Geodetic location and instantaneous speed
  - Geodetic location and instantaneous acceleration – linear or coordinated turn
  - Geodetic location and trajectory type – circular or racetrack
- Vehicle trajectories can be synchronized:
  - User selects multiple vehicle trajectories
  - User selects “Synchronize” from the Range Mission Tool menu
  - User enters synchronization time
- Vehicle RCS is defined:
  - RCS is provided for head-on and side aspect angles

© 2001 Solipsys Corporation

Mission Planning (Vehicles)

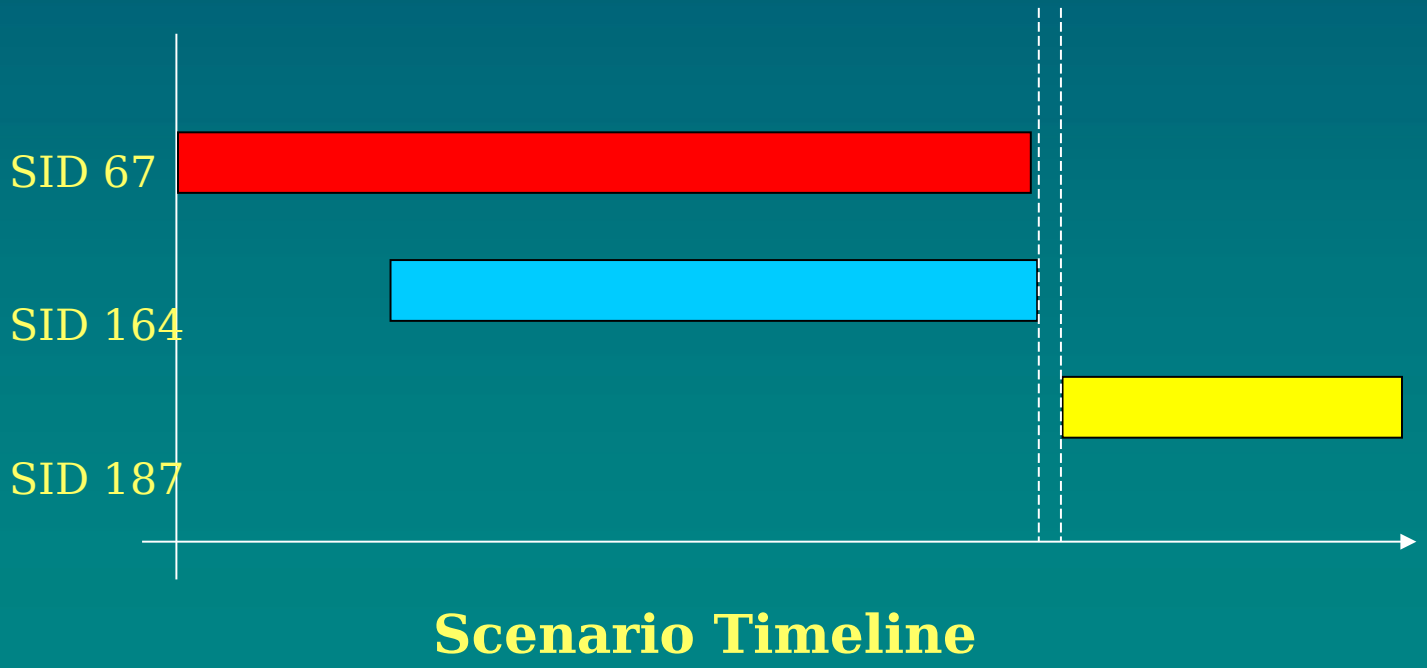
- **S/N Plots:**
  - **A VID is selected by the user**
  - **The S/N Plot option is selected by right-click**
  - **Using sensor model, sensor entity, vehicle waypoints, and vehicle RCS, a S/N graph is generated**



© 2001 Solipsys Corporation

Mission Planning (Vehicle)

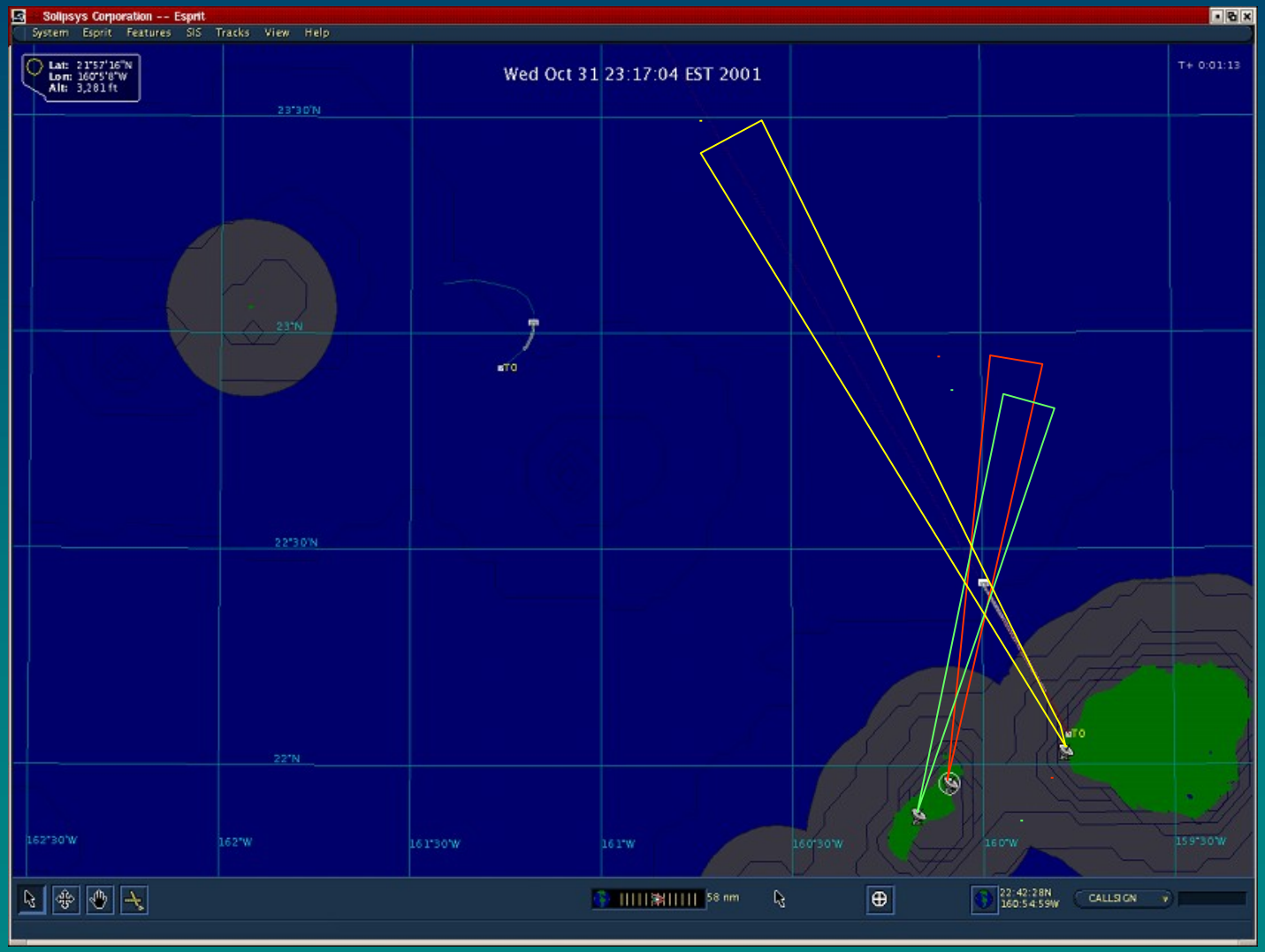
- **Vehicle Coverage Reports:**
  - **User selects VID for radar coverage report**
  - **Using S/N data for this VID, a coverage report is generated for the selected vehicle over the entire scenario timeline**



© 2001 Solipsys Corporation

RANGE MISSION TOOL
SOLIPSYS

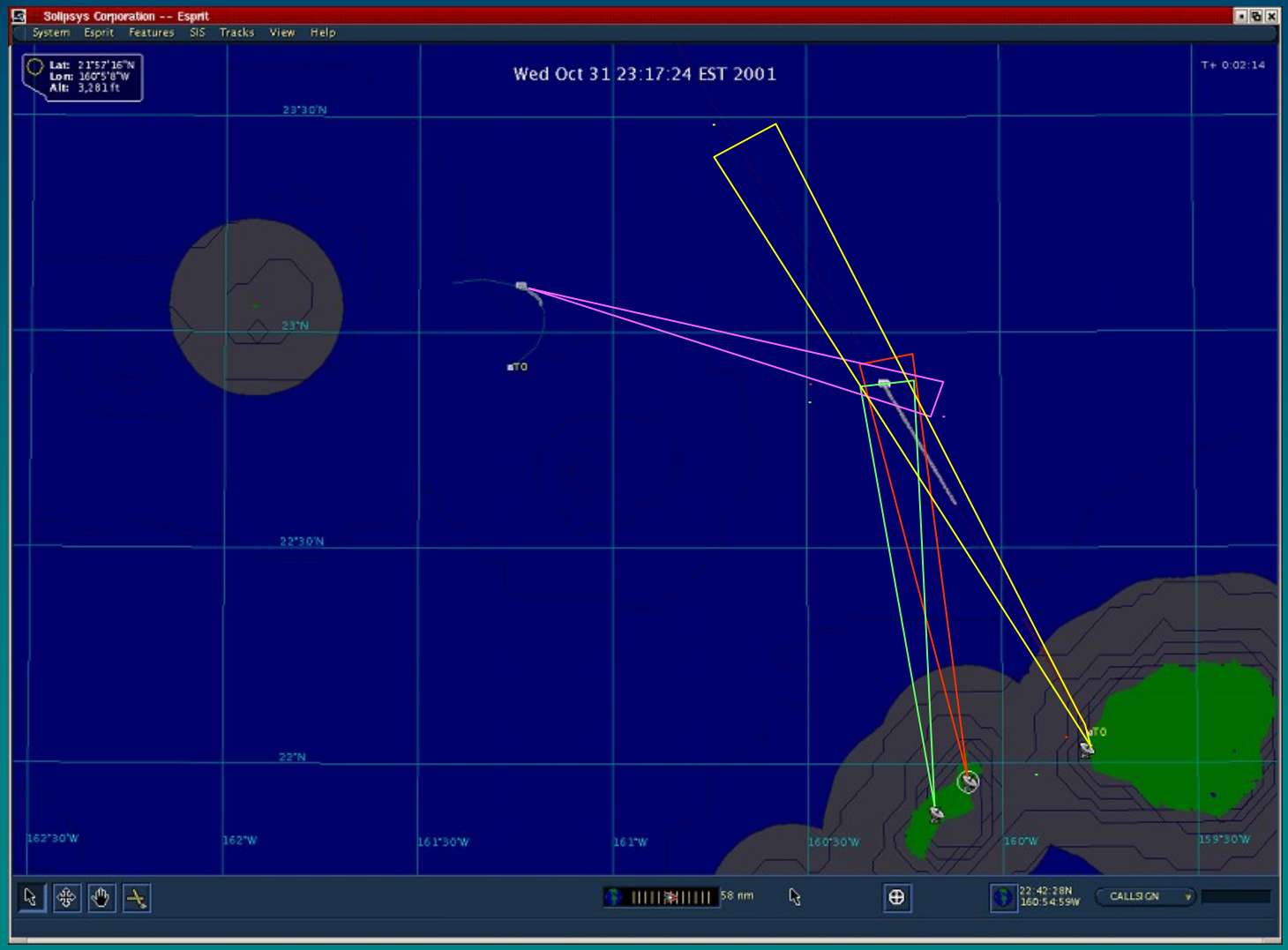
Mission Preview



© 2001 Solipsys Corporation

RANGE MISSION TOOL
SOLIPSYS

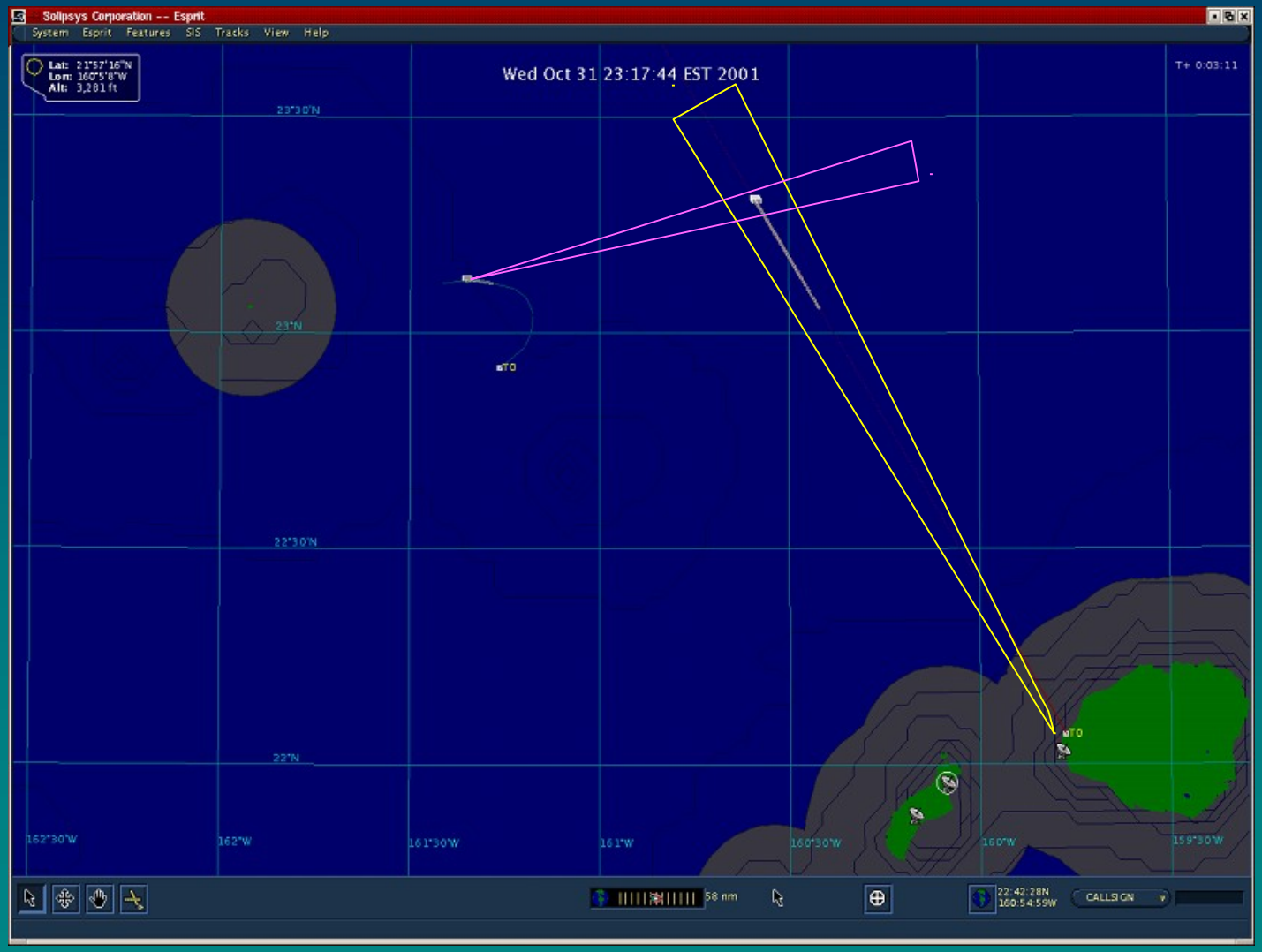
Mission Preview



© 2001 Solipsys Corporation

RANGE MISSION TOOL
SOLIPSYS

Mission Preview



© 2001 Solipsys Corporation

Plan Configuration Management

© 2001 Solipsys Corporation

Scenario Configuration Management

- **Working version**
  - **Used during initial scenario design**
  - **Experimentation encouraged**
- **Certified versions**
  - **Once scenario design work is complete, scenario is certified**
  - **Non-secure password may be employed to restrict certification permission**
  - **Multiple versions of certified scenario need to be saved/accessible**

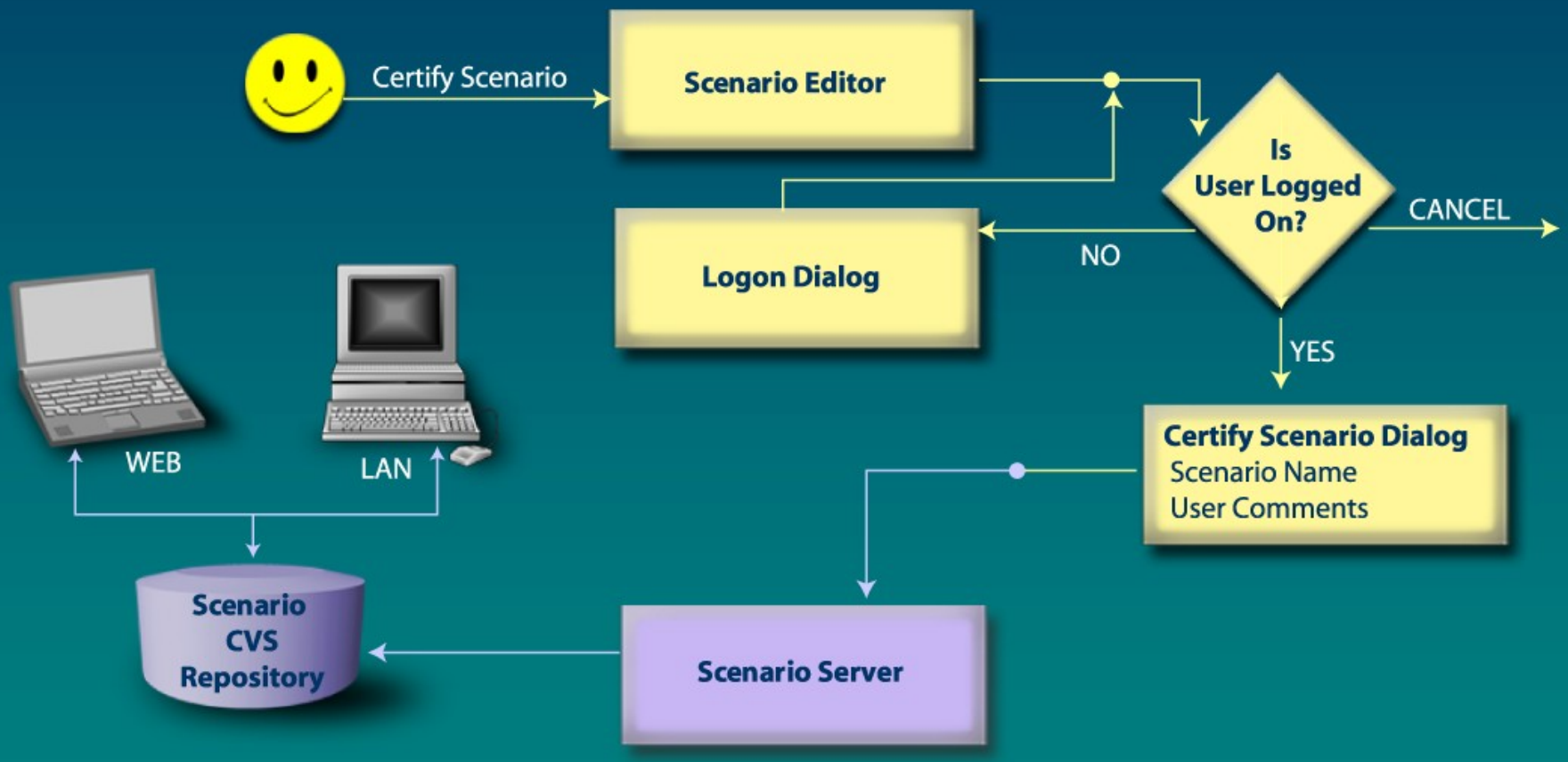
© 2001 Solipsys Corporation

Scenario Configuration Management

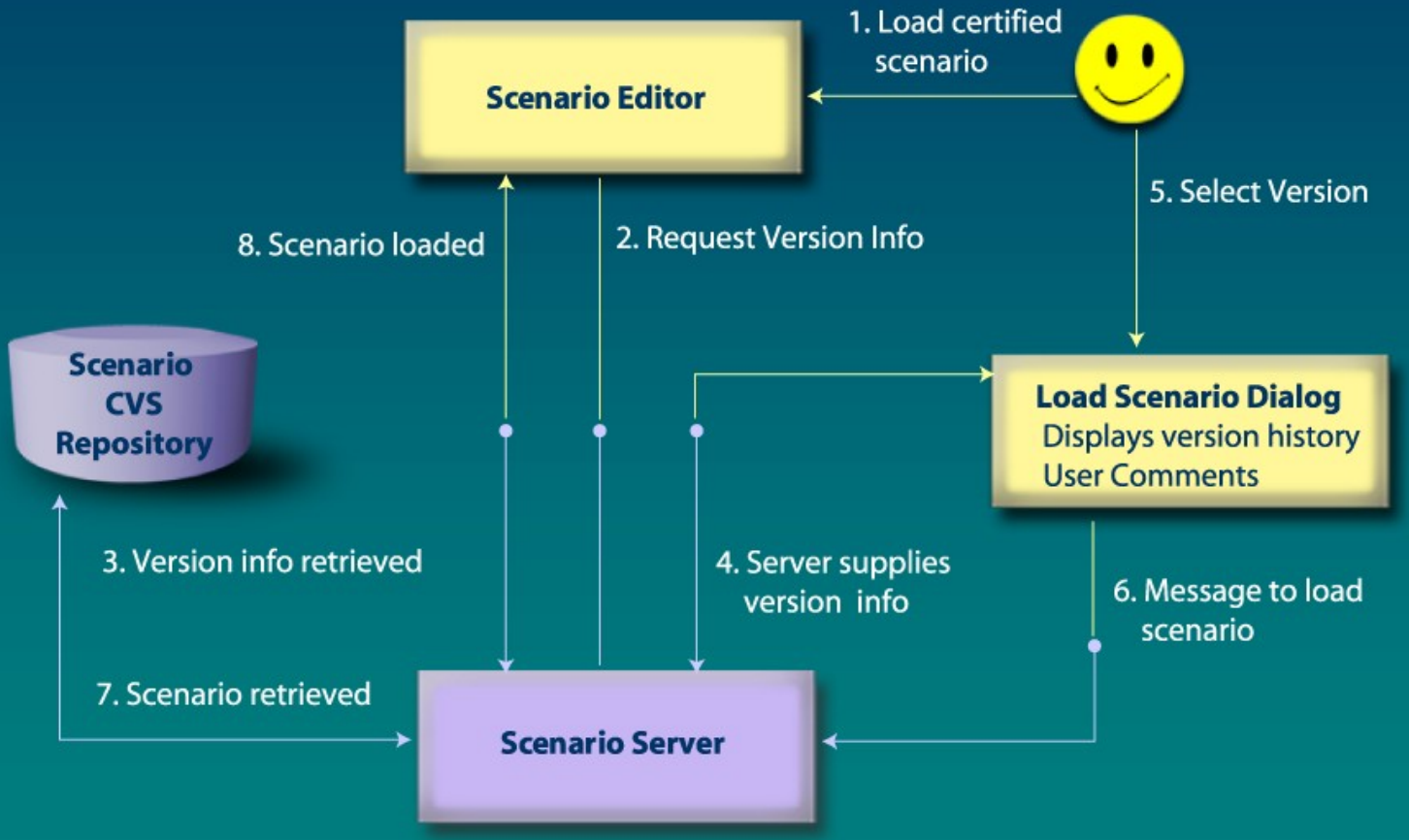
- CVS back end
  - Allows comments to be saved with each certification of a scenario
  - User ID is logged with change
  - Any past versions of a scenario are easily recovered
  - Allows flexible options
    - Shared access from multiple networked sites
    - Web front-end to CVS

© 2001 Solipsys Corporation

Scenario Configuration Management



Scenario Configuration Management



Plan Reports

© 2001 Solipsys Corporation

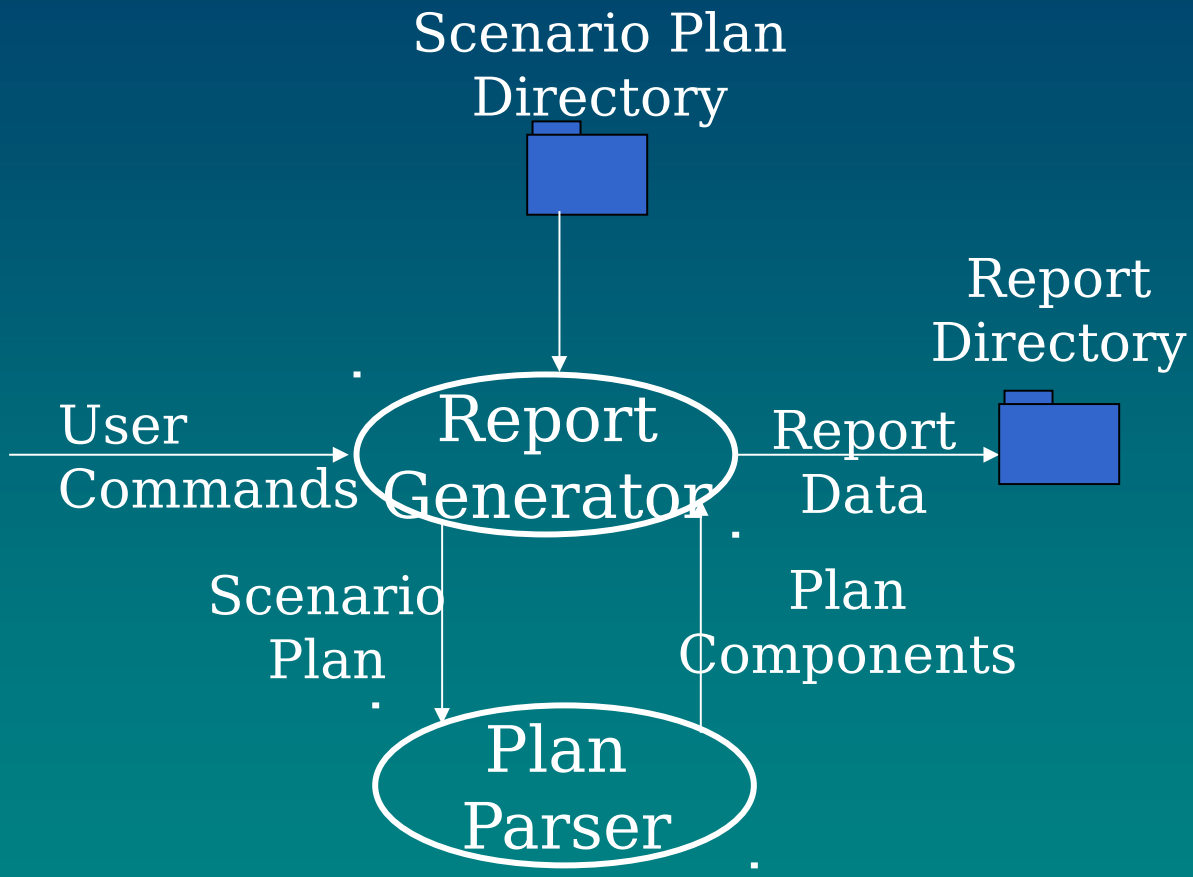
Planning Reports

Plan reports assure consistency and concurrency between the plan and what is published.

- **The Mission Plan is the repository of a considerable amount of mission information**
  - **Vehicle State Report**
  - **Vehicle Timeline Report**
  - **Scenario Timeline Report**
  - **Radar Coverage Report**
  - **Vehicle Information Table (VIT)**
  - **Sensor Information Table (SIT)**

© 2001 Solipsys Corporation

Planning Reports: Data Flow Diagram



© 2001 Solipsys Corporation

Vehicle State Report

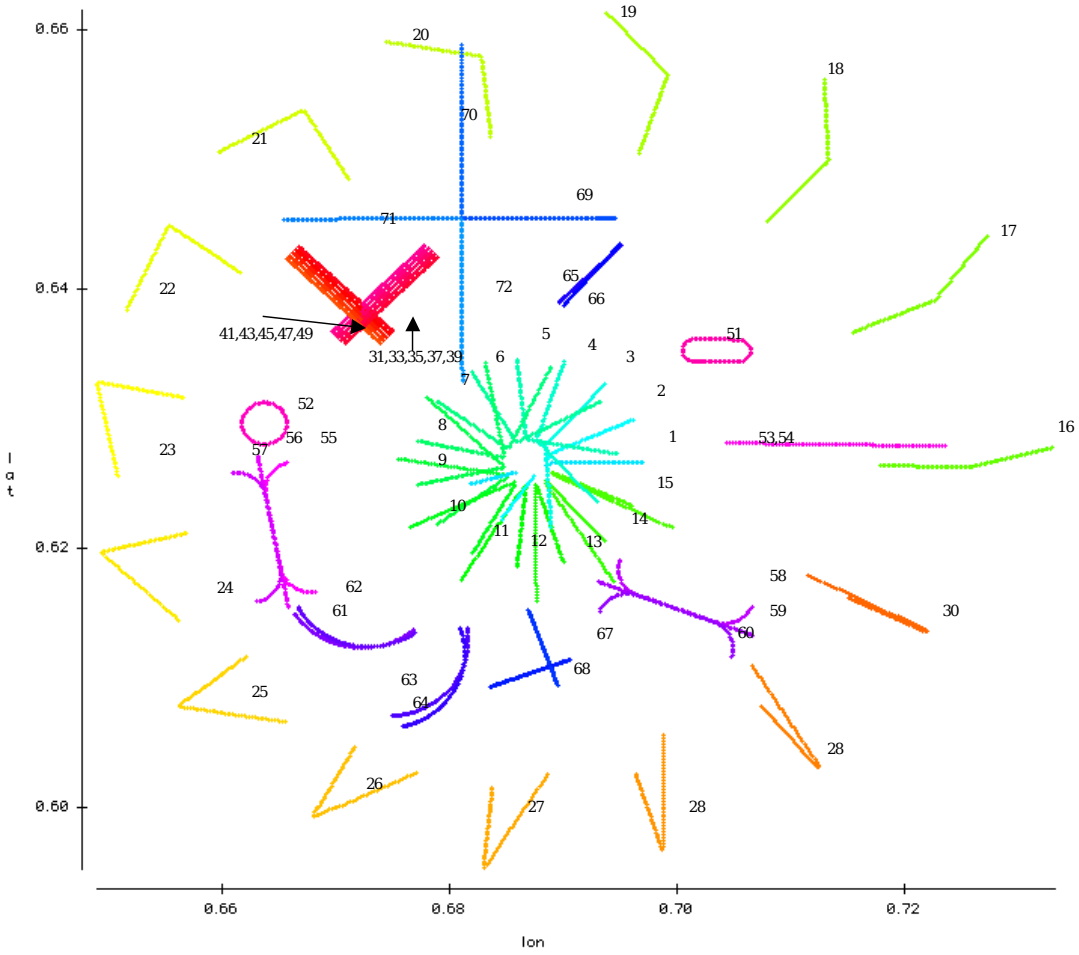
The Vehicle State report provides time, position, velocity and other application pertinent data.

4300 50 1 -89393.15853106 65305.32627878 0.00000000 106.06601718 106.06601718
0.00000000 110591.18244839 -0
.94683067 0.02301077 140.00000000 0.00500000 0.00500000 1 0.62667016 2.25781610
1524.00000000
9350 50 1 -88857.52514431 65840.95966553 0.00000000 106.06601718 106.06601718
0.00000000 110709.81442455 -0
.93785813 0.01382696 140.00000000 0.00500000 0.00500000 1 0.62667016 2.25781610
1524.00000000
14450 50 1 -88316.58845670 66381.89635314 0.00000000 106.06601718 106.06601718
0.00000000 110433.43411118 -
0.92467508 0.01830606 140.00000000 0.00500000 0.00500000 1 0.62667016 2.25781610
1524.00000000
19500 50 1 -87780.95506995 66917.52973989 0.00000000 106.06601718 106.06601718
0.00000000 110610.48018062 -
0.92082443 0.01793941 140.00000000 0.00500000 0.00500000 1 0.62667016 2.25781610
1524.00000000
24600 50 1 -87240.01838234 67458.46642750 0.00000000 106.06601718 106.06601718
0.00000000 110201.84770838 -
0.90692996 0.02354703 140.00000000 0.00500000 0.00500000 1 0.62667016 2.25781610
1524.00000000
29650 50 1 -86704.38499559 67994.09981425 0.00000000 106.06601718 106.06601718
0.00000000 110212.85272888 -
0.90918118 0.02992294 140.00000000 0.00500000 0.00500000 1 0.62667016 2.25781610
1524.00000000

© 2001 Solipsys Corporation

Application of Vehicle State Report

Solipsys application is to use the planning tool to generate realistic test cases for presentation to tracking systems.

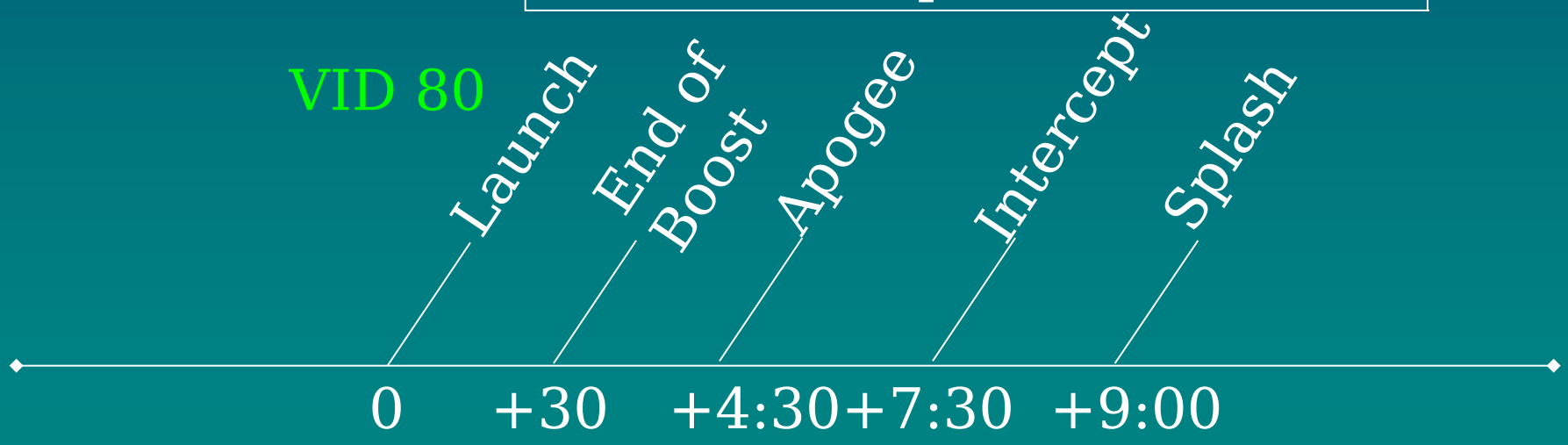


© 2001 Solipsys Corporation

Vehicle Timeline Report

VID 80

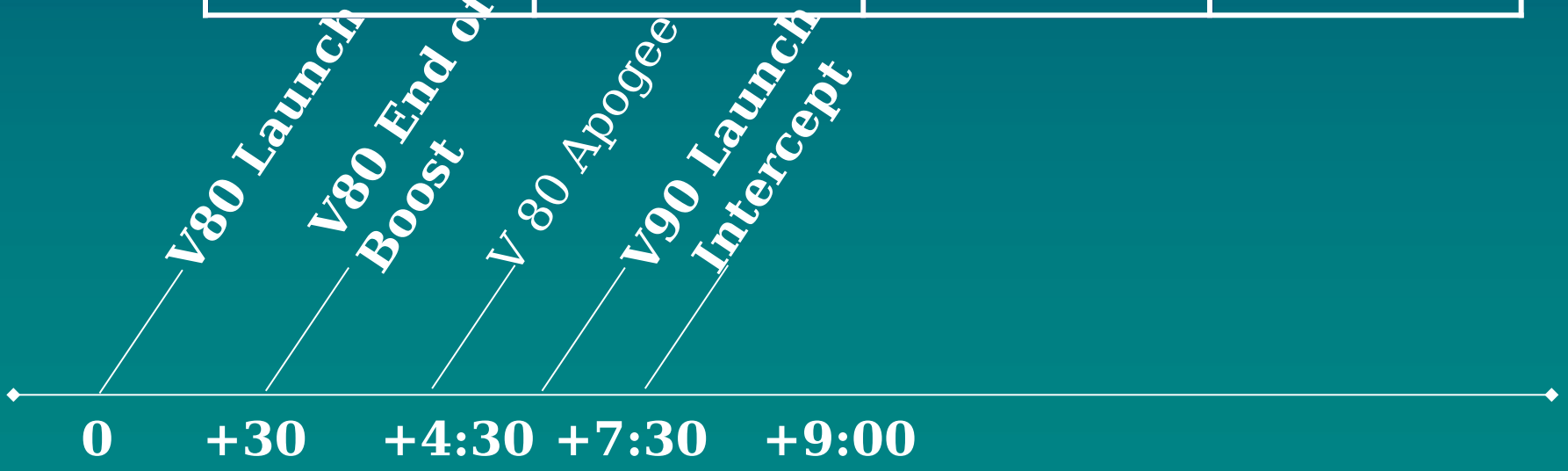
- T+0 Launch
- T+30 End of Boost
- T+4:30 Apogee
- T+7:30 Intercept
- T+9:00 Splash



© 2001 Solipsys Corporation

Scenario Timeline Report

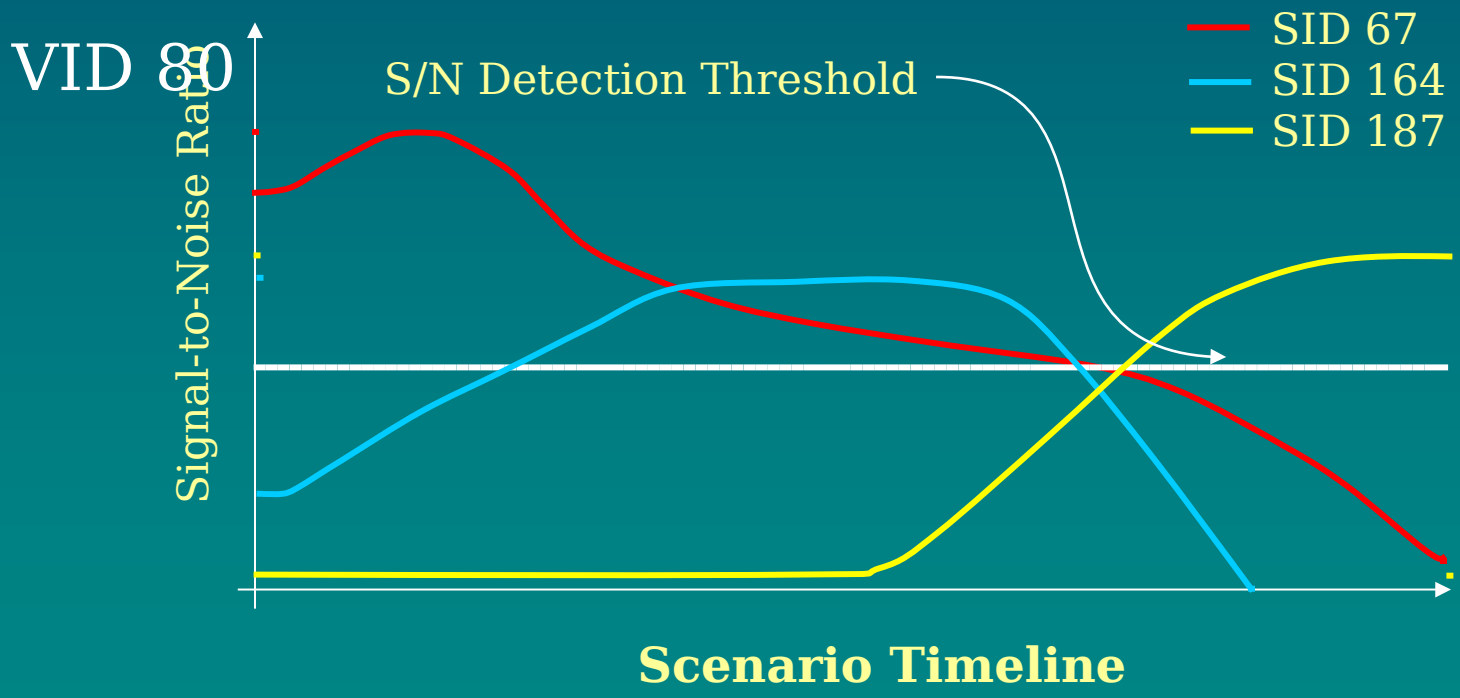
VID	Time	Description	
80	T+0	Launch	
80	+4:30	Apogee	
90	+5:36	Launch	
90	+7:30	Intercept	



© 2001 Solipsys Corporation

Radar Coverage Reports

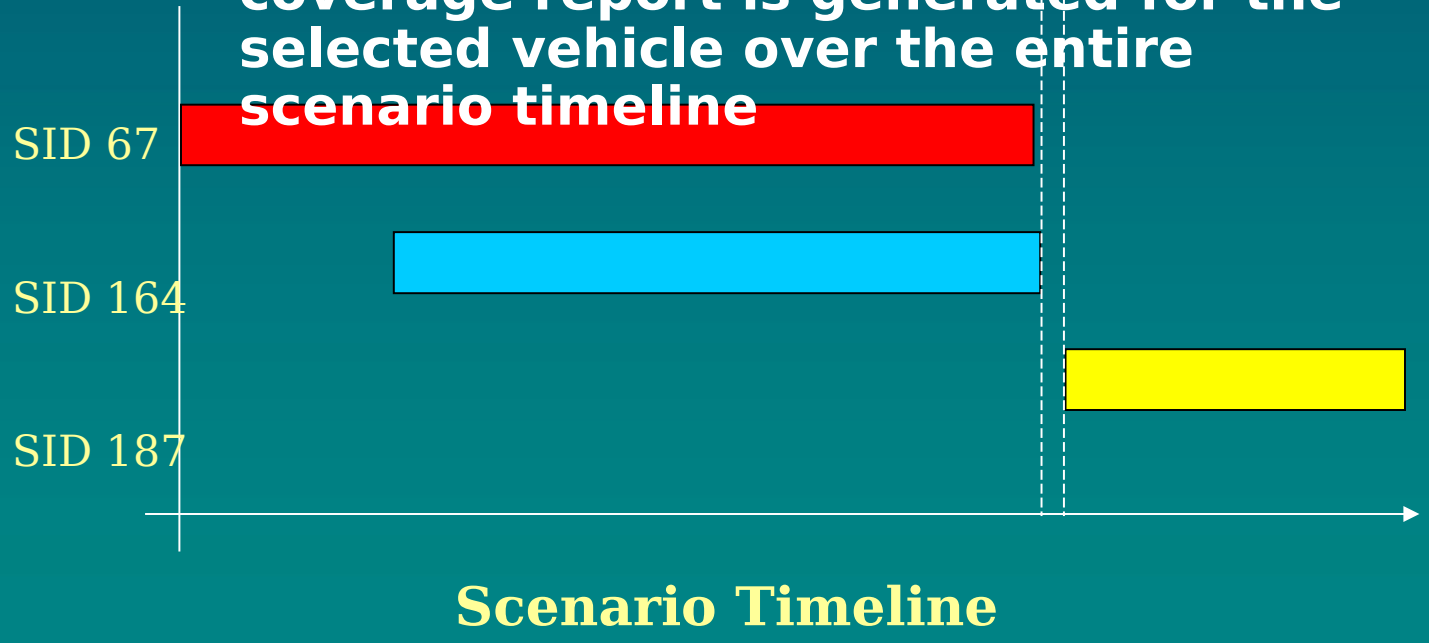
- **S/N Plots:**
  - **Using sensor model, sensor entity, vehicle waypoints and vehicle RCS, a S/N graph is generated**



© 2001 Solipsys Corporation

Radar Coverage Reports

- **Vehicle Coverage Reports:**
  - **User selects VID for radar coverage report**
  - **Using S/N data for this VID, a coverage report is generated for the selected vehicle over the entire scenario timeline**



© 2001 Solipsys Corporation

Vehicle Information Table

VID/SID	Description	Call Sign	Mode 2	Mode 3
23/148	SFIT	USS Anzio	0122	0133
90	SWRB	Weapons Recovery	0432	0239
100	F/A-18A	Fighter Aircraft	0272	0444
870/150	SM-3 TM (VSAT)	Interceptor		
120/220	SM-3 N-STA			

© 2001 Solipsys Corporation

Timeline Enhancements

© 2001 Solipsys Corporation

Vehicle Timeline Synchronization

- Manual Timeline Entry
- Vehicle Timeline Synchronization
- Event Synchronization

Vehicle Trajectory Synchronization allows the planner to express the objectives of the mission.

© 2001 Solipsys Corporation

Manual Timeline Entry

- **Current Capabilities …**
  - **Planner can specify the time (count down time) of any waypoint on a vehicle's trajectory.**
  - **Planner can choose any point of an ASCII data set and designate the point as a time reference point.**
  - **Remainder of the vehicle's trajectory timeline is automatically computed based upon scripted target behavior.**
  - **Time Tic annotations are automatically added in accordance with user specified parameters.**

Current implementation requires all vehicle trajectories be part of the same timeline, i.e. referenced to the same T0.

© 2001 Solipsys Corporation

Vehicle Timeline Synchronization

- **Current Capabilities …**
  - **Planner can select any vehicle waypoint or dataset point and designate it as a synch point.**
  - **Linking synch points from different vehicle trajectories synchronizes the different vehicle timelines**
  - **Any point can be chosen to be the T-zero mark.**
  - **The Mission Tool automatically computes the remainder of each vehicle's timeline based upon vehicle kinematic behavior.**
  - **Time Tic annotations are automatically added in accordance with user specified parameters.**

© 2001 Solipsys Corporation

Event Synchronization

- **Enhanced Capabilities**
  - **Mission Planning Tool shall compute real time events.**
  - **Any vehicle waypoint or dataset point can be synchronized to a computed real time event.**
  - **Detection of the specified event shall initiate/adjust the timelines of all associated vehicles**

Allows multiple independent timelines to be expressed within the same mission plan.

© 2001 Solipsys Corporation